Integrated Adaptive Guidance and Control For Space Access Vehicles

Volume 1 : Reconfigurable Control Law For X-40A Approach and Landing

David B. Doman [1]

Michael W. Oppenheimer [2]

Air Force Research Laboratory, WPAFB, OH 45433-7531

[1]Senior Aerospace Engineer, 2210 Eighth Street, Bldg. 146, Rm. 305, Ph. 937-255-8451, Email David.Doman@wpafb.af.mil

[2]Electronics Engineer, 2210 Eighth Street, Bldg. 146, Rm. 305, Ph. 937-255-8490, Email Michael.Oppenheimer@wpafb.af.mil

Contents

List of Figures

iii

iv

List of Tables

Chapter 1

Introduction

1.1   IAG&C Design Philosophy

This series of reports documents the design, implementation, and testing of an Integrated Adaptive Guidance and Control (IAG&C) System for the X-40A Space Maneuvering Vehicle. This project was motivated by the observation that reconfigurable flight control systems, when used alone, were not always capable of recovering autonomous vehicles following a control effector failure, even though it was physically possible to do so. In manually controlled aircraft, a reconfigurable flight control system can assist pilots in vehicle recovery when failures or damage change the stability and control characteristics. A great deal of work has been accomplished in the area of reconfigurable control systems over the past 30 years [1, 2, 3, 4, 5, 6] and the work presented here builds upon this technological base. While the objective of a reconfigurable control system is to minimize the impact of control effector failures on the handling qualities of the vehicle, one could always allocate the task of trajectory planning to the highly adaptive human pilot. A human pilot can detect degradations in flying qualities in spite of the best efforts of the reconfigurable flight control system (FCS) and still recover the vehicle by planning and executing a flight profile that does not violate the physical capabilities of the impaired system.

In an autonomous flight system, the guidance system and trajectory generation algorithms must fulfill the adaptive role of the pilot in the outer-loop and trajectory planning tasks. One of the objectives of this program was to develop an integrated architecture that enabled a guidance system and tra-

jectory generation algorithm to respond to degradations in closed-inner loop performance resulting from the inability of a reconfigurable FCS to recover nominal performance. Another objective was to create an IAG&C system that could pass a verification and validation process developed by the builder of the X-40A, namely, Boeing Phantom Works. Finally, the IAG&C system was to be flight tested using the USAF/General Dynamics Total In-Flight Simulator (TIFS) aircraft.

In this program, a reconfigurable control law was developed that provided critical information to the guidance and trajectory reshaping algorithms in order to enable these outer-loop and planning systems to account for degraded inner-loop performance due to control effector failures. The baseline control law was designed using Dynamic Inversion in an Explicit Model Following framework. The system is designed to track body-axis pitch, roll, and yaw rate commands. An optimization-based control allocation algorithm is used to produce control effector deflection commands that do not violate actuator rate and position limits. The Onboard Aircraft Model (OAM) made use of multi-dimensional tables to store the force and moment increments from the aerodynamic database. Model bandwidth attenuation and integrator anti-windup algorithms were used to prevent saturation of the body-axis pitch, roll, and yaw rate command systems.

A technique called Reference Model Bandwidth Attenuation (RMBA) was developed that allowed the model following system to account for control power losses and to communicate degraded closed inner-loop performance to the guidance system. RMBA is activated by axis saturation that results from consuming all available rotational acceleration control power in the pitch, roll, or yaw axes. The outputs of the dynamic inversion algorithm, namely, the commanded body angular acceleration vector, serve as the inputs to a linear programming based control allocator. A method for improving the performance of the feedback control/control allocation system was developed that relaxes the typical assumption that the control variable rates are linear functions of the surface deflections with zero intercepts. This new formulation allows a linear control allocator to efficiently and accurately determine control effector positions that satisfy moment or angular acceleration commands when these commands are monotonic and separable functions of control surface deflection. Axis saturation is detected by the control allocator that can compute an instantaneous control deficiency. When a control power deficiency is detected, the bandwidth of the reference model in the explicit model following system is reduced in order to decrease the magnitude

of the angular acceleration commands being passed to the control allocator. Since the angular accelerations are taken to be monotonic functions of control surface deflections, reducing the magnitude of the angular acceleration command decreases the magnitude of the surface deflections, thereby driving them away from saturation. When the closed inner-loop bandwidth is reduced in this way, the gains in the outer-loop guidance system must also be modified to preserve acceptable stability margins.

The adaptive guidance system was developed by Barron Associates and is described in detail in Volume 2 of this report. In essence, the outer-loop gains are adjusted to preserve frequency separation between the inner and outer loops. Also described in Volume 2 is the trajectory generation algorithm. The trajectory generator makes use of a database of trajectories that have been designed off-line for a range of off-nominal conditions. The trajectories are scheduled as a function of states and failure induced constraints. In summary, the fundamental design philosophy is to preserve a modular guidance and control architecture by preventing axis saturation when possible, preserving frequency separation between the inner and outer loops, and generating trajectories for the failed vehicle that minimize control effector demands while enabling the vehicle to land without incurring further damage.

Details of the IAG&C program are provided in this multi-volume set of technical reports, with the current one being Volume 1. In Volume 2, the adaptive guidance and trajectory reshaping algorithms are described, while Volumes 3 and 4 contains details of the Monte-Carlo simulations and generation of flight-ready code performed by Boeing Phantom Works. Lastly, Volume 5 discusses the TIFS aircraft, simulation of the X-40A utilizing the TIFS, and in-flight test results.

Volume 1 is organized as follows: Section 1.2 provides an introduction to the inner-loop reconfigurable control problem. In Section 2.1, the dynamic inversion control law will be discussed. Section 2.2 details the control allocation algorithm, Section 2.3 discusses prefilters for model following, while results are presented in Section 3.1. Conclusions and velocity tracking are discussed in Sections 4.1 and Appendix A, respectively and references are also provided.

1.2 X-40A Background

The X-40A Space Maneuver Vehicle is a technology demonstrator which is an 80% scale version of the X-37 reusable spaceplane. The X-40A Space Maneuver Vehicle (SMV) has a fuselage length of 22 feet, a wing span of 12 feet and weighs about 2,865 pounds. This vehicle is used to validate low-speed handling qualities and demonstrate autonomous approach and landing capabilities. This uniquely shaped vehicle is an innovative approach to fulfilling an Air Force need for a new generation of small and reusable, highly maneuverable space vehicles to perform a variety of tasks such as satellite deployments, surveillance, and logistics missions.

The control design described in this report is for a modified version of the X-40A. The baseline X-40A has at its disposal four control surfaces, left and right flaperons and left and right ruddervators. Unfortunately, these four control surfaces did not provide the level of control redundancy that would be required to fully exercise all elements of the IAG&C system. To alleviate this restriction, two additional control surfaces were augmented to the vehicle model, namely, a speedbrake and a bodyflap. It is important to note that these surfaces exist on the full scale X-37 and would be available on the full-scale re-entry vehicle. In fact, the control increment estimates for the X-40 speedbrake and bodyflap were scaled from the X-37 aerodynamic database by Boeing, as discussed in Volume 3.. Figure 1.1 shows the original X-40A and the location of the additional control effectors. Throughout this report, X-40A will refer to the space maneuvering vehicle with six control surfaces, unless it is specifically stated that the original X-40A (four control surface vehicle) is under consideration. In terms of moment generating capability, the speedbrake and bodyflap essentially produce only pitching moments. Due to their close proximity to the vehicle center of gravity, the flaperons produce very little pitching moment. Hence, with the original X-40A, a failure of the primary pitch control device (ruddervator) would be a catastrophic failure. In the modified X-40A, the vehicle can recover from ruddervator failures because of the additional pitching moment redundancy of the speedbrake and bodyflap.
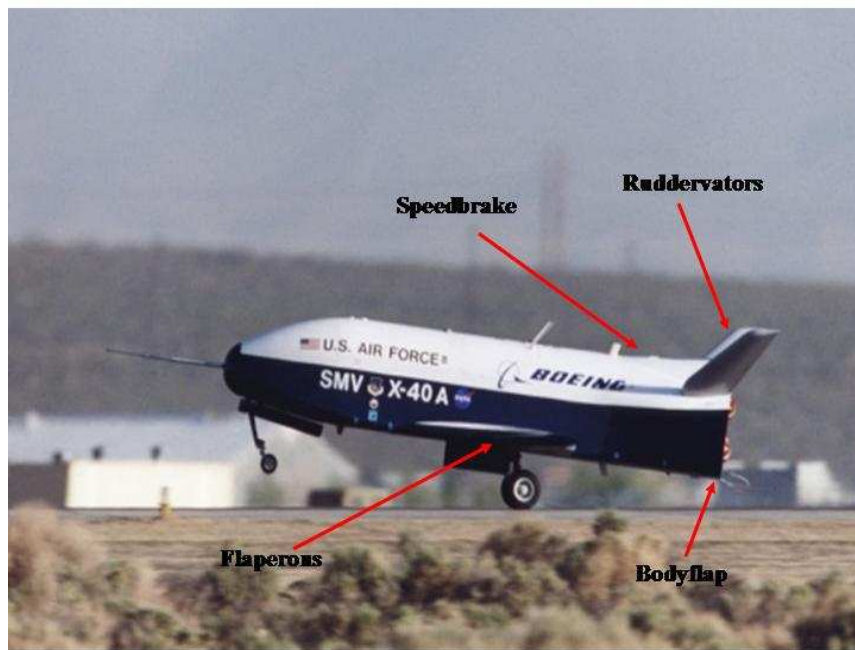
Figure 1.1: X-40A.

Chapter 2

Inner-Loop Reconfigurable Control Design

2.1 Dynamic Inversion

The goal of dynamic inversion in flight control is to cancel the wing-body-propulsion forces and moments with control effector forces and moments such that the vehicle can accurately track some desired commands. Dynamic inversion control laws [6] require the use of a control mixer or control effector allocation algorithm when the number of control effectors exceeds the number of controlled variables, or when actuator rate and position limits must be taken into account. It is quite common that the desired control variable rate commands can be achieved in many different ways and so control allocation algorithms are used to provide unique solutions to such problems [7, 8].

To complete the inner-loop, precompensation blocks are designed to produce the desired closed-loop dynamics. In this work, an explicit model-following prefilter scheme is introduced and inner-loop bandwidth is adjusted by modifying the bandwidth of the explicit model.

For the purpose of demonstration, we develop a dynamic inversion control law for a lifting body reentry vehicle with six control surfaces. The control surfaces include two flaperons, two ruddervators, a speedbrake, and a bodyflap. An outer-loop control system generates body-frame angular velocity commands $(p_{des}, q_{des}, r_{des})$, that the inner-loop dynamic inversion control system attempts to track. The development of the vehicle model for the angular velocities follows the work of Doman, et. al. [9]. The dynamics of

the body-frame angular velocity vector for a lifting body can be written as

$$\dot{\boldsymbol{\omega}} = \mathbf{f}(\boldsymbol{\omega}, \mathbf{P}) + \mathbf{g}(\mathbf{P}, \boldsymbol{\delta}) \tag{2.1}$$

where $\boldsymbol{\omega} = [\text{p q r}]^T$, and p, q, and r are the rolling, pitching, and yawing rates, respectively, $\mathbf{P}$ denotes measurable or estimable quantities that influence the body-frame states, and $\boldsymbol{\delta} = (\delta_1, \delta_2, \cdots, \delta_n)^T$ is a vector of control surface deflections. The vector $\mathbf{P}$ contains variables such as angle of attack, sideslip, Mach number, and vehicle mass properties. The term $\mathbf{g}(\mathbf{P}, \boldsymbol{\delta})$ includes the control dependent accelerations, while the term $\mathbf{f}(\boldsymbol{\omega}, \mathbf{P})$ describes accelerations that are due to the base-vehicle's (wing-body-propulsion) aerodynamic properties. The moment equations for a lifting body in the body-frame [10] can be manipulated to form control dependent and control independent terms. It is assumed that the mass properties of the vehicle under consideration are constant, thus, the time derivative of the inertia matrix can be set to zero, i.e., $\dot{\mathbf{I}} = \mathbf{0}$. Then, Equation 2.1 can be written as

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(\mathbf{G}_B(\boldsymbol{\omega}, \mathbf{P}, \boldsymbol{\delta}) - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}) \tag{2.2}$$

where

$$\mathbf{G}_B(\boldsymbol{\omega}, \mathbf{P}, \boldsymbol{\delta}) = \mathbf{G}_{BAE}(\boldsymbol{\omega}, \mathbf{P}) + \mathbf{G}_\delta(\mathbf{P}, \boldsymbol{\delta}) = \begin{bmatrix} L \\ M \\ N \end{bmatrix}_{BAE} + \begin{bmatrix} L \\ M \\ N \end{bmatrix}_\delta \tag{2.3}$$

In Equations 2.2 and 2.3, $\mathbf{I}$ is the inertia matrix and L, M, and N are the rolling, pitching, and yawing moments. In Equation 2.3, $\mathbf{G}_{BAE}(\boldsymbol{\omega}, \mathbf{P})$ is the moment generated by the base aerodynamic system (wing-body-propulsion system) and $\mathbf{G}_\delta(\mathbf{P}, \boldsymbol{\delta})$ is the sum of moments produced by the control effectors. Therefore,

$$\begin{aligned} \mathbf{f}(\boldsymbol{\omega}, \mathbf{P}) &= \mathbf{I}^{-1}[\mathbf{G}_{\mathbf{BAE}}(\boldsymbol{\omega}, \mathbf{P}) - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}] \\ \mathbf{g}(\mathbf{P}, \boldsymbol{\delta}) &= \mathbf{I}^{-1}\mathbf{G}_\delta(\mathbf{P}, \boldsymbol{\delta}) \end{aligned} \tag{2.4}$$

In order to utilize a linear control allocator, it is necessary that the control dependent portion of the model be linear in the controls. Hence, an affine approximation is developed such that

$$\mathbf{G}_\delta(\mathbf{P}, \boldsymbol{\delta}) \approx \widetilde{\mathbf{G}}_\delta(\mathbf{P})\boldsymbol{\delta} + \boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta}) \tag{2.5}$$

The term $\boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta})$ is an intercept term [11] for the body-axis angular accelerations which is used to improve the accuracy of linear control allocation

7

algorithms. More detail on this term will discussed in subsequent sections. Using Equations 2.1, 2.4, and 2.5, the model used for the design of the dynamic inversion control law becomes

$$\dot{\boldsymbol{\omega}} = \mathbf{f}(\boldsymbol{\omega}, \mathbf{P}) + \mathbf{I}^{-1}\widetilde{\mathbf{G}}_\delta(\mathbf{P})\boldsymbol{\delta} + \mathbf{I}^{-1}\boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta}) \tag{2.6}$$

The objective is to find a control law, that provides direct control over $\dot{\boldsymbol{\omega}}$, so that $\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_{des}$. Hence, the inverse control law must satisfy

$$\dot{\boldsymbol{\omega}}_{des} - \mathbf{f}(\boldsymbol{\omega}, \mathbf{P}) - \mathbf{I}^{-1}\boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta}) = \mathbf{I}^{-1}\widetilde{\mathbf{G}}_\delta(\mathbf{P})\boldsymbol{\delta} \tag{2.7}$$

Equation 2.7 provides the dynamic inversion control law for the body-frame angular velocity vector.

2.2   Control Allocation

Since there are more control effectors than controlled variables and the control effectors are restricted by position and rate limits, a control allocation algorithm must be used. For the lifting body under consideration, there are three controlled variables, namely, roll, pitch, and yaw rates, while there are six control surfaces (left and right flaperons, left and right ruddervators, speedbrake, and bodyflap). Hence, a control allocation scheme must be used to insure that Equation 2.7 is satisfied. The control allocation scheme used in this work draws heavily on the work of Buffington, et. al. [12, 13].

Control allocators are used in conjunction with some type of feedback control law whose output consists of one or more pseudo-control commands (typically desired moment or acceleration commands). The number of pseudo-control commands is always less than or equal to the number of control effectors. Dynamic inversion control laws and control allocation algorithms fit together quite naturally since the pseudo-control commands are easily identifiable. Also, it is quite common that the desired commands can be achieved in many different ways and so control allocation algorithms are used to provide unique solutions to such problems.

To begin development of the allocator, consider rewriting Equation 2.7 as

$$\mathbf{d}_{des} = \dot{\boldsymbol{\omega}}_{des} - \mathbf{f}(\boldsymbol{\omega}, \mathbf{P}) - \mathbf{I}^{-1}\boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta}) = \mathbf{I}^{-1}\widetilde{\mathbf{G}}_\delta(\mathbf{P})\boldsymbol{\delta} = \mathbf{B}\boldsymbol{\delta} \tag{2.8}$$

where $\mathbf{d}_{des}$ are the body-axis accelerations that must be produced by the control effectors and $\mathbf{B}$ is the control effectiveness matrix defined as

$$\mathbf{B} = \mathbf{I}^{-1}\widetilde{\mathbf{G}}_\delta(\mathbf{P}) = \mathbf{I}^{-1} \begin{bmatrix} \frac{\partial L}{\partial \delta_1} \frac{\partial L}{\partial \delta_2} \cdots \frac{\partial L}{\partial \delta_n} \\ \\ \frac{\partial M}{\partial \delta_1} \frac{\partial M}{\partial \delta_2} \cdots \frac{\partial M}{\partial \delta_n} \\ \\ \frac{\partial N}{\partial \delta_1} \frac{\partial N}{\partial \delta_2} \cdots \frac{\partial N}{\partial \delta_n} \end{bmatrix} \tag{2.9}$$

The control allocation objective, in the linear case, is to find $\boldsymbol{\delta}$ such that

$$\mathbf{d}_{des} = \mathbf{B}\boldsymbol{\delta} \tag{2.10}$$

subject to rate and position limits on the control effectors. Notice that Equation 2.10 defines a linear subspace in the $(\mathbf{d}_{des}, \boldsymbol{\delta})$ space, a point which will be utilized in Section 2.2.2.

Equation 2.10 can be posed as the following optimization problem:

$$\min_\delta J_E = \min_\delta \|\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}\|_1 \tag{2.11}$$

subject to

$$\overline{\boldsymbol{\delta}} \leq \boldsymbol{\delta} \leq \underline{\boldsymbol{\delta}} \tag{2.12}$$

where J_E is the performance index for the error minimization problem, $\underline{\boldsymbol{\delta}}$, $\overline{\boldsymbol{\delta}}$ are the most restrictive lower and upper limits on the control effectors, respectively and the 1-norm is selected so that linear programming techniques can be used to solve the problem [12]. More specifically,

$$\begin{aligned} \overline{\boldsymbol{\delta}} &= \min(\boldsymbol{\delta}_U, \boldsymbol{\delta} + \dot{\boldsymbol{\delta}}_{max}\Delta t) \\ \underline{\boldsymbol{\delta}} &= \max(\boldsymbol{\delta}_L, \boldsymbol{\delta} - \dot{\boldsymbol{\delta}}_{max}\Delta t) \end{aligned} \tag{2.13}$$

where $\boldsymbol{\delta}_L, \boldsymbol{\delta}_U$ are the lower and upper position limits, $\boldsymbol{\delta}$ is the current location of the control effectors, $\dot{\boldsymbol{\delta}}_{max}$ is a vector of rate limits, and Δt is the timestep or control update rate.

If sufficient control authority exists such that J_E can be made identically equal to zero, then it may be possible to optimize a sub-objective. This optimization problem can be posed as follows:

$$\min_\delta J_C = \min_\delta \|\mathbf{W}_\delta(\boldsymbol{\delta} - \boldsymbol{\delta}_p)\|_1 \tag{2.14}$$

9

subject to

$$\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des}$$
$$\overline{\boldsymbol{\delta}} = \min(\boldsymbol{\delta}_U, \boldsymbol{\delta} + \dot{\boldsymbol{\delta}}_{max}\Delta t) \qquad (2.15)$$
$$\underline{\boldsymbol{\delta}} = \max(\boldsymbol{\delta}_L, \boldsymbol{\delta} - \dot{\boldsymbol{\delta}}_{max}\Delta t)$$

where $\mathbf{W}_\delta$ is a weighting matrix and $\boldsymbol{\delta}_p$ is a preferred set of control effector deflections. The problem posed in Equation 2.14 is termed the control minimization problem.

In practice, the two optimization problems given in Equations 2.11 and 2.14 are combined to form what is known as the mixed optimization problem [8]. The mixed optimization problem is defined as

$$\min_\delta J_M = \min_\delta \left(\|\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}\|_1 + \lambda \|\mathbf{W}_\delta(\boldsymbol{\delta} - \boldsymbol{\delta}_p)\|_1 \right) \qquad (2.16)$$

where the parameter λ is used to weight the error and control minimization problems. For this work, it was determined that $\lambda = 0.01$ provided good error minimization while still driving the control effectors to the preferred values when sufficient control authority existed. The advantage of the mixed optimization problem is that it can often be solved faster and with better numerical properties as compared to sequentially solving the error and control minimization problems [8].

2.2.1 Control Allocation Preference Vector and Effector Failures

As specified in Equation 2.16, a preference vector, $\boldsymbol{\delta}_p$, must be selected. One difficulty with the linear programming framework for solving the control allocation problem is that no closed-form model of the control allocator exists. This causes problems when performing linear stability analysis as there is no way of representing the input/output relationship of the allocation algorithm. Fortunately, when sufficient control authority exists, the allocation algorithm can attempt to minimize the difference between the control deflections and a preferred set of control deflections. One obvious choice for preference vector is the pseudo-inverse solution or minimum 2-norm solution. In this case, when sufficient control authority exists, the control allocation algorithm will drive the surfaces to the pseudo-inverse solution. Hence, in a robustness analysis, the control allocator can be replaced by the closed-form solution

(assuming sufficient control authority exists). The pseudo-inverse solution is the minimum two-norm solution to the control allocation problem and can be formulated as follows:

$$\min_{\boldsymbol{\delta}} \frac{1}{2}(\boldsymbol{\delta} + \mathbf{c})^{\mathbf{T}}\mathbf{W}(\boldsymbol{\delta} + \mathbf{c}) \tag{2.17}$$

subject to

$$\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des} \tag{2.18}$$

where $\mathbf{W}$ is a weighting matrix and $\mathbf{c}$ is an offset vector. To solve this problem, first find the Hamiltonian (H) such that

$$H = \frac{1}{2}\boldsymbol{\delta}^T\mathbf{W}\boldsymbol{\delta} + \frac{1}{2}\mathbf{c}^T\mathbf{W}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T\mathbf{W}\mathbf{c} + \frac{1}{2}\mathbf{c}^T\mathbf{W}\mathbf{c} + \boldsymbol{\xi}(\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}) \tag{2.19}$$

where $\boldsymbol{\xi} \in \mathbb{R}^n$ is an as yet undetermined vector of Lagrange multipliers. Taking the partial derivatives of H with respect to $\boldsymbol{\delta}$ and $\boldsymbol{\xi}$, setting these expressions equal to zero, and rearranging, gives

$$\frac{\partial H}{\partial \boldsymbol{\delta}} = \mathbf{W}\boldsymbol{\delta} + \frac{1}{2}(\mathbf{c}^{\mathbf{T}}\mathbf{W})^T + \frac{1}{2}\mathbf{W}\mathbf{c} + (\boldsymbol{\xi}\mathbf{B})^T = \mathbf{0}$$
$$\implies \mathbf{W}\boldsymbol{\delta} = -\mathbf{W}\mathbf{c} - \mathbf{B}^{\mathbf{T}}\boldsymbol{\xi}^{\mathbf{T}} \tag{2.20}$$

and

$$\frac{\partial H}{\partial \boldsymbol{\xi}} = \mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des} = \mathbf{0}$$
$$\implies \mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des}$$
$$\implies \mathbf{B}\mathbf{W}^{-1}\mathbf{W}\boldsymbol{\delta} = \mathbf{d}_{des} \tag{2.21}$$

Substituting Equation 2.20 into Equation 2.21 yields

$$\mathbf{B}\mathbf{W}^{-1}[-\mathbf{W}\mathbf{c} - \mathbf{B}^T\boldsymbol{\xi}^T] = \mathbf{d}_{des} \tag{2.22}$$

Solving for $\boldsymbol{\xi}^T$ in Equation 2.22 yields

$$\boldsymbol{\xi}^T = -(\mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T)^{-1}[\mathbf{d}_{des} + \mathbf{B}\mathbf{c}] \tag{2.23}$$

Substituting Equation 2.23 into Equation 2.20 produces

$$\mathbf{W}\boldsymbol{\delta} = -\mathbf{W}\mathbf{c} + \mathbf{B}^T(\mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T)^{-1}[\mathbf{d}_{des} + \mathbf{B}\mathbf{c}] \tag{2.24}$$

Simplifying Equation 2.24 gives the desired result

$$\boldsymbol{\delta} = \boldsymbol{\delta}_p = -\mathbf{c} + \mathbf{W}^{-1}\mathbf{B}^T(\mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T)^{-1}[\mathbf{d}_{des} + \mathbf{B}\mathbf{c}] \qquad (2.25)$$

Equation 2.25 gives the pseudo-inverse solution. It should be noted that if an effector is offset, two items must be taken into account, position offset $(-\mathbf{c})$ and the moments generated by the offset $(\mathbf{B}\mathbf{c})$. For the specific usage of the pseudo-inverse control allocation solution, the weighting matrix was selected to be diagonal, such that,

$$\mathbf{W} = diag\left[W_{\delta_{RF}}\ W_{\delta_{LF}}\ W_{\delta_{RR}}\ W_{\delta_{LR}}\ W_{\delta_{SB}}\ W_{\delta_{BF}}\right] \qquad (2.26)$$

where 'diag' represents a diagonal matrix with the entries along the main diagonal being the weights associated with each control effector.

This control allocation formulation allows one to simulate a control effector failure rather easily. A failure is introduced by simply setting the lower and upper positions limits on the effected control surface equal to each other. For a failed control surface, its effects must also be accounted for in the pseudo-inverse preference vector, which requires two modifications. First, the location of the failure must be inserted into the offset vector. Here, the appropriate component of $\mathbf{c}$ is set to the negative of the failure position. Second, the appropriate entry in the weighting matrix, $\mathbf{W}$, must be increased. Nominally, the entries in $\mathbf{W}$ are one and an increase in the value will place more penalty on usage of that particular surface.

2.2.2 Improving Control Allocation Accuracy

The primary function of the control allocation algorithm is, given the current control effectiveness matrix ($\mathbf{B}$) and a vector of desired pseudo-control commands ($\mathbf{d}_{des}$), find the control effector settings $\boldsymbol{\delta}$ that will produce the desired pseudo-control commands. A method is presented that improves the accuracy of linear control allocators when the moments produced by the effectors are monotonic and separable nonlinear functions of effector position [11]. Hence, it is desired to improve the accuracy of conventional allocation algorithms. The modification developed in the upcoming paragraphs will be described by use of a one-dimensional example, although this theory is directly applicable to multiple dimensions. It is important to note that the improvement in allocation algorithms is measured in terms of the accuracy of the calculation of $\boldsymbol{\delta}$.

Most control allocation algorithms use the assumption that a linear relationship exists between the pseudo-control-commands ($\mathbf{d}_{des}$) and the control effector displacements ($\boldsymbol{\delta}$), i.e.,

$$\mathbf{d}_{des} = \mathbf{B}\boldsymbol{\delta} \tag{2.27}$$

where $\mathbf{B}$ is the control effectiveness matrix which, in terms of Equation A.3, is defined as

$$\mathbf{B} = \mathbf{I}^{-1}\widetilde{\mathbf{G}}_\delta(\mathbf{P}) \tag{2.28}$$

This is equivalent to a linear subspace containing the zero element. In the one-dimensional case, Equation 2.27 describes a line passing through the origin in the d_{des}, δ plane. In reality, the forces and moments produced by aerodynamic control surfaces are often nonlinear functions of control surface deflection. This phenomenon limits the accuracy of linear control allocation algorithms since most currently available approaches are based on the assumption that the control variable rates are linear functions of the surface deflections and that the control variable rate increments are not produced for zero deflections. The errors introduced by this assumption are currently mitigated by the robustness resulting from feedback control laws. A method for improving the performance of the feedback control/control allocation system, that directly attacks the inaccuracies introduced by the linear assumptions, was developed for this program.

To begin, consider the one-dimensional example shown in Figure 2.1. Here, a typical moment-deflection curve is shown with the dashed curve. Control effector position is indicated on the horizontal-axis, and moment produced by the effector is indicated on the vertical-axis. At time t_k, the control effector is currently at δ_k and is producing a moment given by $g_\delta(P_o, \delta_k)$. At time t_{k+1}, assume that the desired moment is $g_\delta(P_o, \delta_{k+1})$ as shown in Figure 2.1. In order to produce the desired moment, the control effector must be in the position given by $\delta_{k+1}^{desired}$. This value is obtained by drawing a horizontal line from $g_\delta(P_o, \delta_{k+1})$ to the moment curve (dashed curve), then dropping a vertical line to the δ axis. If the moment-deflection model used by the allocation algorithm accurately modelled the moment-deflection relationship, then it would ideally calculate $\delta_{k+1}^{desired}$ as the required control effector setting at time t_{k+1}. Unfortunately, the only information that the linear control allocation algorithm has in regards to the moment curve is a slope (global slope in this example). If a horizontal line is drawn from $g_\delta(P_o, \delta_{k+1})$ to the global slope line and a vertical line is drawn from this intersection to the δ axis, it is
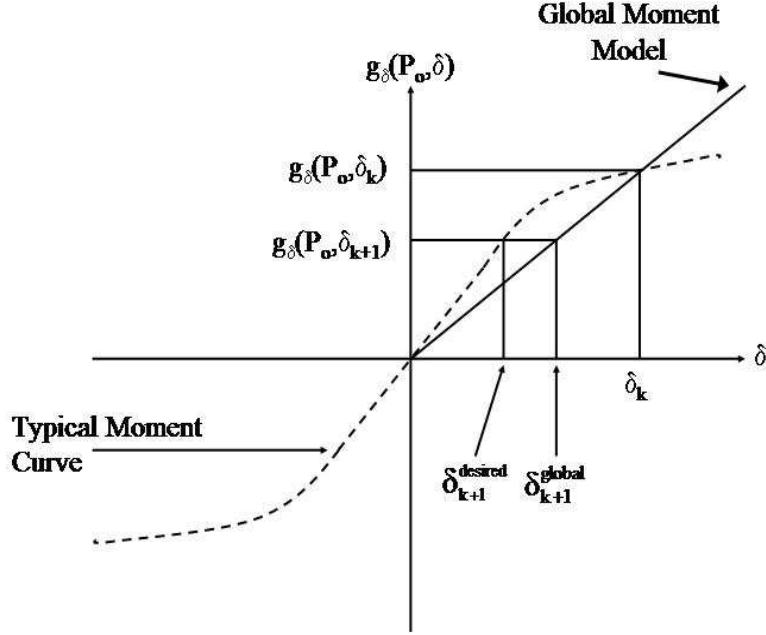
Figure 2.1: Effects of Global Slope Model on Control Allocation Algorithm.

found that the linear control allocation algorithm will compute δ_{k+1}^{global} as the required control effector setting. Thus, it is easily seen that there can be a large discrepancy between the desired (perfect knowledge of control effectiveness) and actual (local slope knowledge of control effectiveness). It should be pointed out that nonlinear control allocation algorithms can be used to precisely compute δ_{k+1}, however, it is our experience that they cannot be utilized in practice since they cannot complete the required computations in the available amount of time [14]. The computational restrictions of current flight computers have forced the use of linear control allocation algorithms.

In practice, the pseudo-controls are nonlinear functions of the surface deflections as well as other parameters ($\mathbf{P}$) that typically include angle-of-attack, sideslip, and Mach number. Therefore, the pseudo-controls can be represented by

$$\mathbf{d}_{des} = \mathbf{g}(\mathbf{P}, \boldsymbol{\delta}) \tag{2.29}$$

where $\mathbf{g}(\mathbf{P}, \boldsymbol{\delta})$ is some nonlinear function of parameters and controls. At any

14

given flight condition and deflection, one could find a $\mathbf{B}$ for which $\mathbf{B}\boldsymbol{\delta} = \mathbf{g}(\mathbf{P}, \boldsymbol{\delta})$; however, the individual control effectiveness parameters (slopes) would be inaccurate.

An improvement to the accuracy of state-of-the-art control allocators, while not violating the linear assumptions, can be obtained by including an intercept term. This intercept term [11], $\boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta})$, is used to adjust the pseudo-control commands. In other words, we make use of an approximation for $\mathbf{g}(\mathbf{P}, \boldsymbol{\delta})$ of the form

$$\mathbf{d}_{des} = \mathbf{g}(\mathbf{P}, \boldsymbol{\delta}) = \mathbf{B}\boldsymbol{\delta} + \mathbf{I}^{-1}\boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta}) \tag{2.30}$$

Then, a new pseudo-control command $\mathbf{d}'_{des}$ is formed such that

$$\mathbf{d}'_{des} = \mathbf{d}_{des} - \mathbf{I}^{-1}\boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta}) = \mathbf{B}\boldsymbol{\delta} \tag{2.31}$$

On-line, the intercept term can be computed using the following:

$$\mathbf{I}^{-1}\boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta}) = \mathbf{I}^{-1} \begin{bmatrix} L \\ M \\ N \end{bmatrix}_{\delta} - \mathbf{B}\boldsymbol{\delta} \tag{2.32}$$

Figure 2.2 shows a block diagram displaying the dynamic inversion control law, control allocator, plant, and intercept correction term.

In order to describe the effects of $\boldsymbol{\epsilon}(\mathbf{P}, \boldsymbol{\delta})$ on the calculation of $\boldsymbol{\delta}_{k+1}$, consider another one-dimensional example. Shown in Figure 2.3 is a moment curve (dashed curve) and two current operating conditions, δ_1 and δ_2. Contrary to the case displayed in Figure 2.1, in which a global slope approximation was used, linear moment models (local slopes) are used in this case. It can also be seen that, given the operating conditions, these local slope lines intersect the $g_\delta(P_o, \delta)$ axis at nonzero locations. These nonzero locations are the intercept correction terms, $\epsilon_1(P_o, \delta_1)$ and $\epsilon_2(P_o, \delta_2)$.

Having illustrated $\epsilon(P_o, \delta)$, its impact on the control allocation algorithm will now be formulated. Figure 2.4 shows the effect of $\epsilon(P_o, \delta)$ on the calculation of δ_{k+1}. Shown in Figure 2.4 is again a one-dimensional representation of a moment curve along with a current operating condition given by δ_k. Also shown are locally linear (accurate control effectiveness) and global (inaccurate control effectiveness) slope models at the current operating condition, and the nonzero intercept, $\epsilon_1(P_o, \delta_k)$, for the local slope. At time t_k, the moment $g_\delta(P_o, \delta_k)$ is being produced. Assume, at time t_{k+1}, the desired moment

15

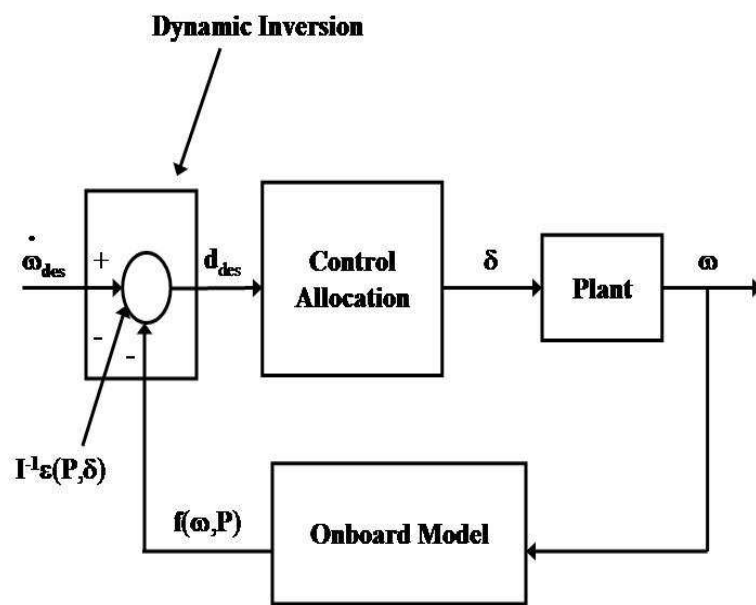Figure 2.2: Inner-Loop Dynamic Inversion with Intercept Correction.

to be produced is $g_\delta(P_o, \delta_{k+1})$. Drawing a horizontal line from $g_\delta(P_o, \delta_{k+1})$ through the two approximate slopes and the moment curve then dropping vertical lines to the δ axis results in three points. First is the correct value of δ_{k+1}. If the control effector were at this location, the exact desired moment, $g_\delta(P_o, \delta_{k+1})$, would be produced. With the local with intercept and global slope approximations, the computed value of δ_{k+1} is different than the correct value. However, it is easily seen that utilizing the local slope with intercept results in a better approximation for δ_{k+1}.

The same type of behavior is displayed when considering using the local slope with and without intercept correction. In this case, both of the control effectiveness parameters are accurate. Figure 2.5 displays this situation. Once again, the one-dimensional moment curve is displayed, the current operating condition is given by δ_k, and there are now two linear control moment models with slopes equal to the local slope. The first linear control model has a zero $g_\delta(P_o, \delta)$ intercept and represents a global slope model that is based on the local slope of the actual moment-deflection curve. In this case, the control effectiveness is correct (local slope), however, the additional intercept information is not used. The second linear control model has a nonzero $g_\delta(P_o, \delta)$ intercept and the control effectiveness is correct (local slope). Again, of interest here is the effect of $\epsilon_1(P_o, \delta_k)$ on the calculation of δ_{k+1}. At time t_{k+1}, assume that the desired moment to be produced is $g_\delta(P_o, \delta_{k+1})$. Using the same logic as in Figure 2.4, it can be seen that the accuracy of the calculation of δ_{k+1} is improved when the local slope with intercept is used as compared to the global approximation to the local slope (local slope with no intercept).

One last point to consider in the control allocation paradigm is the potential for the existence of slope reversals in the moment-deflection curve data. On the X-40A, slope reversals sometimes exist near the maximum or minimum values of control surface deflection. Figure 2.6 shows the scenario in which the moment curve is non-monotonic. A problem can arise when the actuator moves into the non-monotonic region because the sign of the control effectiveness parameter changes. Thus, when a smaller moment is commanded, the effector will actually move towards the upper limit δ_{max}. Actually, the desired response is for the effector to move "over-the-hump" in the moment curve back to a region where $\delta < \delta_{peak}$. Without proper precautions, these slope reversals can cause an effector to be confined to a region given by $([\delta_{peak}, \delta_{max}])$. For linear control allocation algorithms, the non-monotonic region can be clipped to ensure that the moment curve is
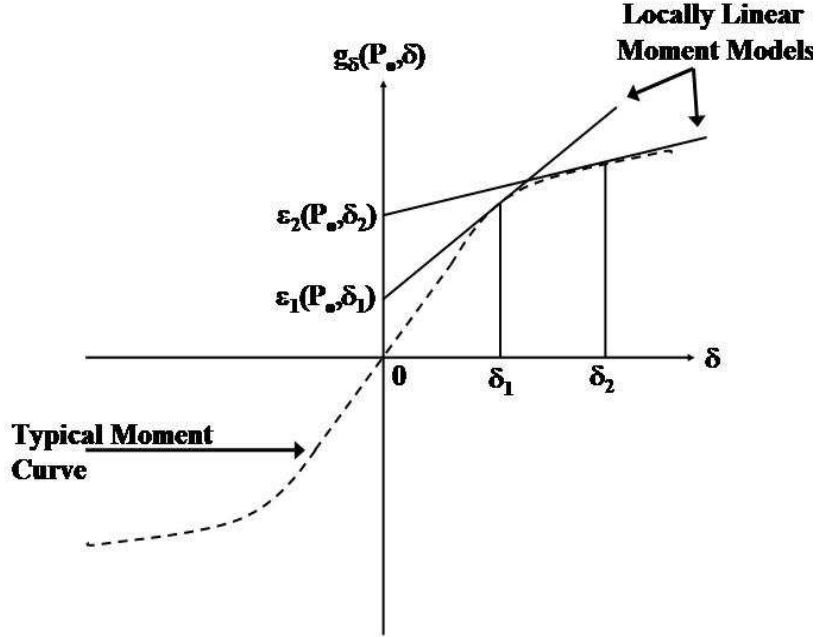
Figure 2.3: Moment Curve With Intercept Correction.

monotonic throughout the control space, as shown in Figure 2.7. In this way, the problems encountered with slope reversals are eliminated. The clipping is achieved by forcing the moment-deflection relationship to be a horizontal line once the non-monotonic region is reached. Hence, further deflection of the control effector produces no change in moment, with respect to the control allocation algorithm. The actual aerodynamic data used in the dynamic inversion control law is not modified in this way, only the data used to determine the effectiveness of the control effector (entries in the $\mathbf{B}$ matrix). It is also important to point out that the clipping is only performed for slope reversals near the upper or lower bounds of the control effector. If a slope reversal occurs at other locations, the data is not modified.
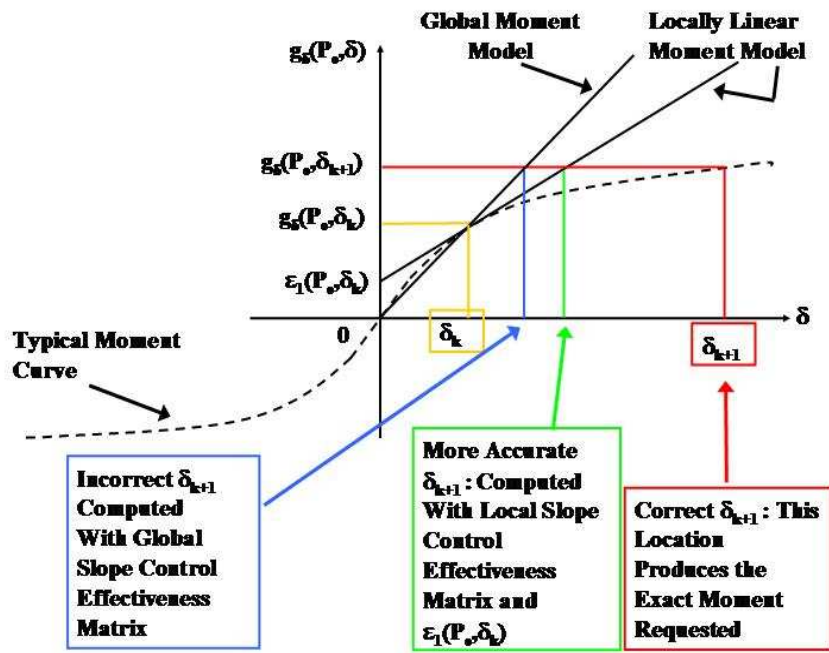
18

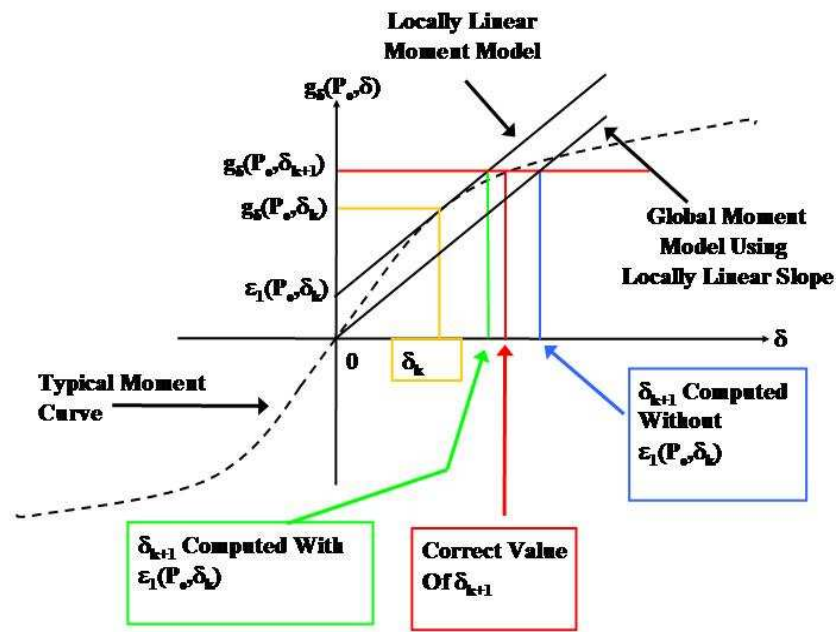Figure 2.4: Effect of Intercept Correction on Calculation of δ.

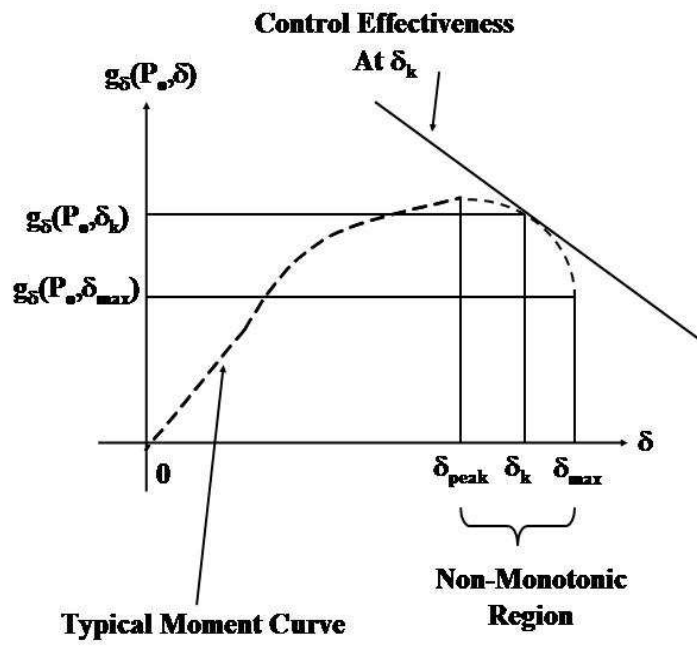Figure 2.5: Effect of Intercept Correction on Calculation of δ.

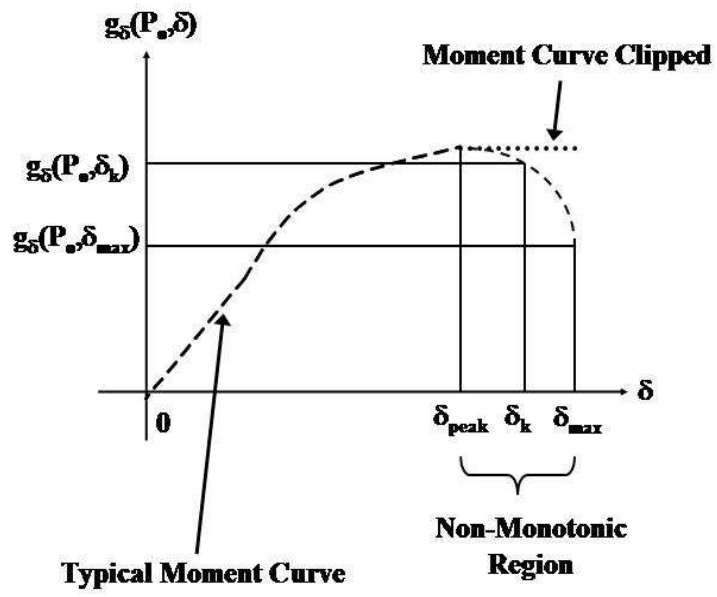Figure 2.6: Non-Monotonic Moment Curve With Intercept Correction.

Figure 2.7: Clipping of Non-Monotonic Moment Curve.

2.3 Explicit Model Following

The inner-loop flight control system was designed so that the closed-inner-loop system would exhibit a decoupled first order response to body-axis angular rate commands. An explicit model following scheme was used to shape the closed-inner-loop response and to compensate for imperfections in the dynamic inversion control law. Explicit models of the desired roll, pitch and yaw dynamics were captured in the following framework:

$$\frac{\omega_m(s)}{\omega_{cmd}(s)} = \frac{K_{bw}}{s + K_{bw}} \tag{2.33}$$

where ω_m denotes either the desired roll, pitch, or yaw rate response of the explicit model and ω_{cmd} denotes the angular velocity command from the guidance and control interface. The term K_{bw} defines the nominal bandwidth of the desired dynamics. Figure 2.8 shows a block diagram of a single channel of an explicit model following structure that motivated the design of the system implemented in this work. Practical considerations preventing the direct implementation of this design include the requirement of discrete-time operation, axis saturation, and sensor noise. Nevertheless it is a useful tool to establish the basic design of the model following system. The system is designed to provide perfect tracking of the reference model when the dynamic inversion is perfect. Since this is never the case in practice, error compensation elements are used to mitigate the effects of inversion error. If the inversion is perfect, then the controlled element from the point of view of the explicit model following structure is a simple integrator. From block diagram algebra, one can see that if the controlled element is a simple integrator, the $\omega(s)/\omega_m(s)$ transfer function is given by:

$$\frac{\omega(s)}{\omega_m(s)} = \frac{(K_D + K_{FF})s^2 + K_P s + K_I}{(K_D + 1)\, s^2 + K_P s + K_I} \tag{2.34}$$

which, when $K_{FF} = 1$, results in a double stable pole-zero cancellation with appropriate choices of K_P, K_I, and K_D. Then, it is easily seen that

$$\frac{\omega(s)}{\omega_{cmd}(s)} = \frac{K_{bw}}{s + K_{bw}} \tag{2.35}$$

and perfect model following is achieved.

When the inversion is not perfect, the PID network attempts to drive the reference model tracking error to zero. Table 2.1 shows the gains for each

channel of the explicit model following system used in the TIFS/X-40A flight tests. These gains were selected to provide acceptable tracking performance for pulse train commands in all channels.

Since the control system must operate in discrete time, z-transform representations of continuous time elements were used. Tustin transformations were used to convert continuous time elements into discrete time elements. The discrete time elements are shown in Figure 2.9. Performing a Tustin transformation on the reference model results in:

$$\frac{\omega_m(s)}{\omega_{cmd}(s)} = \frac{K_{bw}}{s + K_{bw}}$$

$$\Rightarrow \frac{\omega_m(z)}{\omega_{cmd}(z)} = \frac{K_{bw}T(z-1)}{(K_{bw}T+2)z + (K_{bw}T-2)} \tag{2.36}$$

The "derivative" channel makes use of a discrete time representation of a first order high-pass filter with a 20 rad/sec break frequency to avoid amplification of high frequency noise, i.e.,

$$\frac{D(s)}{e(s)} = \frac{20K_D s}{s + 20}$$

$$\Rightarrow \frac{D(z)}{e(z)} = \frac{16.67K_D(z-1)}{z - .6667} \tag{2.37}$$

The integrator channel provides for windup protection and uses a Tustin transformation to convert a continuous time integrator to a discrete time representation, i.e.,

$$I(s) = \frac{K_I}{s}(e(s) - I_{AW}(s))$$

$$\Rightarrow I(z) = \frac{T(z+1)}{2(z-1)}K_I(e(z) - I_{AW}(z)) \tag{2.38}$$

The anti-windup signal I_{AW} is used to prevent the integrator from compensating for steady-state errors that are the result of axis saturation. Details are given in the following section.

2.3.1 Integrator Anti-Windup and Reference Model Bandwidth Attenuation

Control effector saturation results when one or more control surfaces is moving at its rate limit or lies on a position limit. Axis saturation occurs when
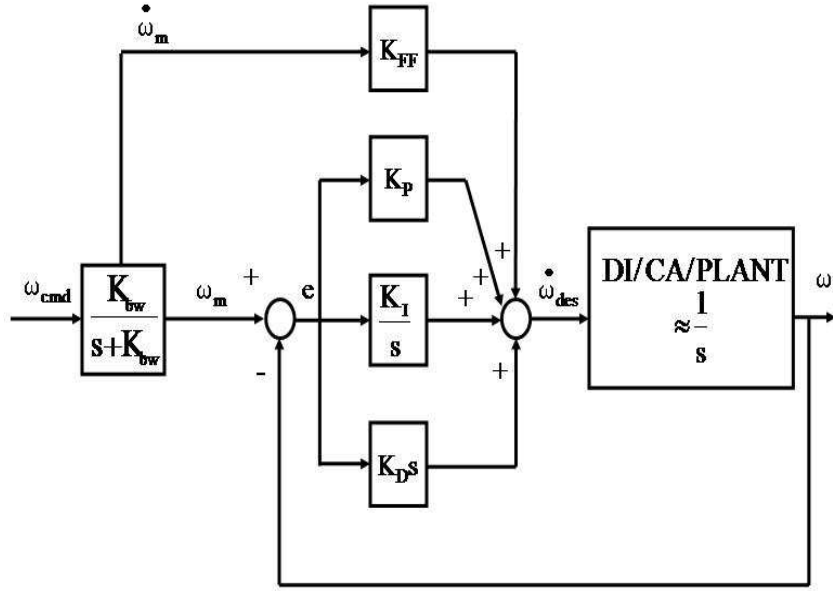
Figure 2.8: Block Diagram of Single Channel of Conceptual Explicit Model Following Control System.

Roll	Pitch	Yaw
$K_{bw_p} = 5$	$K_{bw_q} = 3$	$K_{bw_r} = 5$
$K_{P_p} = 7$	$K_{P_q} = 10$	$K_{P_r} = 7$
$K_{I_p} = 20$	$K_{I_q} = 10$	$K_{I_r} = 20$
$K_{D_p} = .1$	$K_{D_q} = .1$	$K_{D_r} = .1$
$K_{FF_p} = 1$	$K_{FF_q} = 1$	$K_{FF_r} = 1$

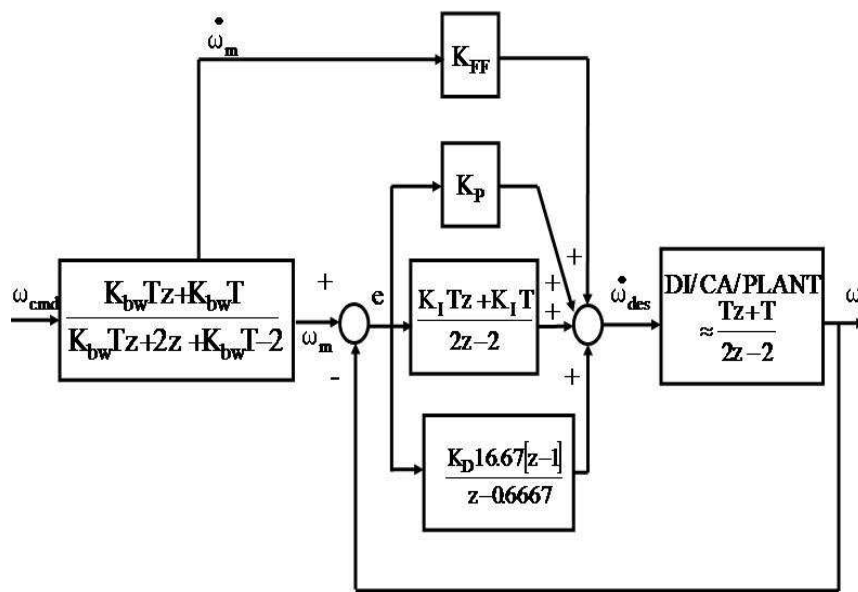Table 2.1: Control system parameters for explicit model following control system.

Figure 2.9: Block Diagram of Single Channel of Discrete Explicit Model Following Control System with Integrator Anti-Windup Protection.
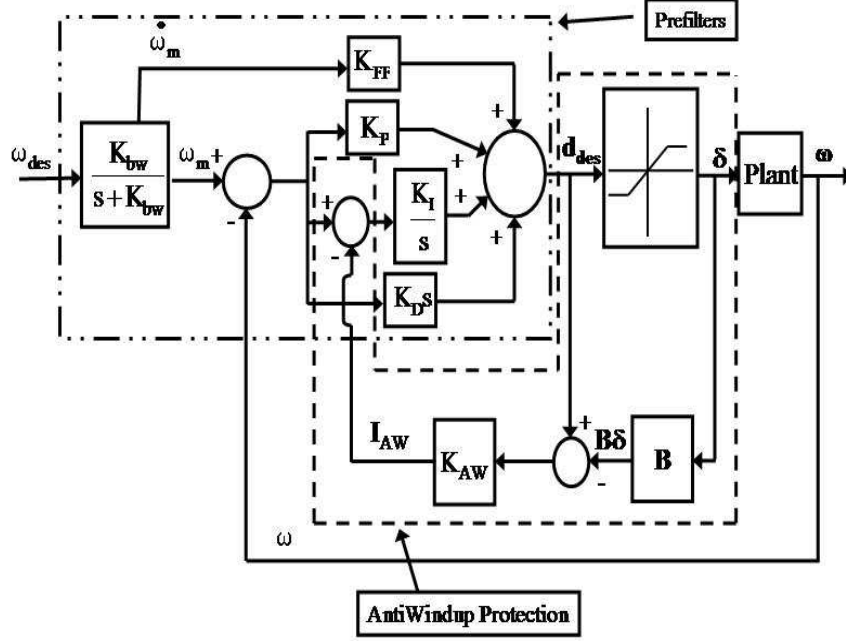
26

Figure 2.10: Block Diagram of Explicit Model Following Prefilter Integrator Anti-Windup Compensation.

control power has been depleted in one or more axes. For this program, control effector saturation is a necessary, but not sufficient, condition for the occurrence of axis saturation. Axis saturation can be detected through an analysis of the control allocation inputs and outputs. If $\mathbf{B}\boldsymbol{\delta}_{cmd} - \mathbf{d}_{des} \neq \mathbf{0}$ then axis saturation has occurred. In order to prevent the integrator in the explicit model following prefilter from attempting to cancel tracking errors caused by axis saturation, an integrator anti-windup law is used to reduce the magnitude of input to the integrator. The integrator anti-windup vector used in this design is given by:

$$\mathbf{I}_{AW} = \mathbf{K}_{AW}(\mathbf{B}\boldsymbol{\delta}_{cmd} - \mathbf{d}_{des}) \tag{2.39}$$

where $\mathbf{K}_{AW}$ is a gain. The block diagram of the explicit model following prefilter integrator antiwindup scheme is shown in Figure 2.10.

The integrator antiwindup compensation scheme operates on the difference between the pseudo control command (angular acceleration vector com-

mand), $\mathbf{d}_{des}$, and the output of the control effectiveness model used by the control allocator, $\mathbf{B}\boldsymbol{\delta}$. If no axes are saturated, then $\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des} = \mathbf{0}$ and the control system operates normally. When $\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des} \neq \mathbf{0}$, at least one axis is saturated and the state of the prefilter integrators are reduced by the anti-windup signals.

When an axis saturates, that is, when at least one component of $\mathbf{B}\boldsymbol{\delta}_{cmd} - \mathbf{d}_{des} \neq \mathbf{0}$, the inner-loop control system's performance is degraded and most likely will not be able to track the nominal commands. One way of conveying this information to an outer-loop guidance system is by way of inner-loop bandwidth. Nominally, the inner-loop bandwidth is set to $\mathbf{K}_{bw}$, however, when axis saturation occurs, this bandwidth should be reduced to avoid over-driving the actuators. Figure 2.11 shows the scheme used to reduce the bandwidth ($\mathbf{K}_{bw}$) of the explicit models whenever axis saturation occurs. Take, for example, the roll (P) channel shown in Figure 2.11. If $(\mathbf{B}_P\boldsymbol{\delta} - \mathbf{d}_{des_P}) = \mathbf{0}$, then the roll loop continues to operate at its nominal bandwidth. Here, $\mathbf{B}_P$ is the row of the $\mathbf{B}$ matrix corresponding to the roll axis and d_{des_P} is the body-axis roll acceleration that is desired from the control effectors. For situations where $(\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}) \neq \mathbf{0}$, a method was devised to decrease the bandwidth depending on the magnitude of the difference between $\mathbf{B}_i\boldsymbol{\delta}$ and $\mathbf{d}_{des_i}, i = 1, 2, 3$. The saturation blocks restrict the inputs to the gain blocks to be between 0 and 1, inclusive. Using the bandwidth attenuation logic of Figure 2.11, a large error would result in a small bandwidth (minimum bandwidth is 0.5 rad/sec for all three axes). Reducing the bandwidth after sensing saturation of one or more axes has the effect of reducing the commands to the control effectors thereby reducing the tendency to saturate.

The inner-loop control system is shown in Figure 2.12. This figure displays the prefilters, integrator antiwindup protections, dynamic inversion, control allocation, the plant, and the inter-connections of all these components. This is the complete IAG&C inner-loop control system for the X-40A.
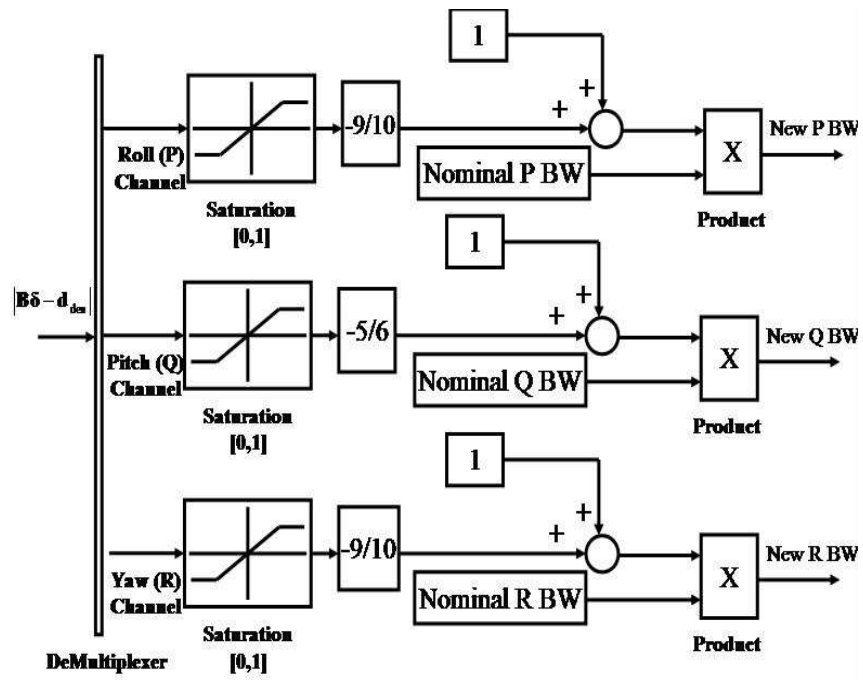
Figure 2.11: Inner-Loop Bandwidth Modification Scheme.

Figure 2.12: Inner-Loop Control System.

Chapter 3

Results

3.1   Simulation

A six degree-of-freedom simulation was developed to test the algorithms and concepts developed in this work. The vehicle was commanded to follow a nominal trajectory from the reentry point to the touchdown point.

A set of five representative simulation runs will be shown here. These include a nominal case and four failure cases. The failure cases investigated include a right flaperon locked at 15°, a bodyflap locked at 5°, and a combination failure consisting of the left and right ruddervators locked at 0°. Each of these cases provide unique information that will be discussed in the next sections.

For each failure case, the results consist of the following items:

1. Command and Actual Roll, Pitch, and Yaw Rates from Simulation

   Displays roll, pitch, and yaw rate commands and the actual roll, pitch, and yaw rates produced by the vehicle. Shows the tracking performance of the inner-loop.

2. Actual and Ideal Roll, Pitch, and Yaw Rates from Simulation

   Shows the actual roll, pitch, and yaw rates produced by the vehicle and the roll, pitch, and yaw rates produced by an ideal representation of the inner-loop (first-order transfer function). Note that the roll, pitch, and yaw rate simulation commands drive both the actual rates and the ideal rates responses. Displays the performance of the control allocator

and dynamic inversion since if these are perfect, then the inner-loop would be equivalent to a first-order transfer function. Small differences in these traces can be attributed to errors produced by the control allocation algorithm. Errors cannot be attributed to the dynamic inversion control law during simulation since the model used by the dynamic inversion controller is the truth model and no disturbances are present.

3. Control Effector Deflections from Simulation

4. Inner-Loop Bandwidth (if applicable) from Simulation

   Inner-loop bandwidth is modified when one or more axes saturate. For some of the failure cases investigated in this report, no axes saturate and it is therefore not necessary to modify the inner-loop bandwidth. When inner-loop bandwidth is altered, this is an indication that all control power has been expended in one or more axes.

5. Altitude versus Downrange from Simulation

6. Altitude, Crossrange, and Downrange from Simulation

7. Actual and Ideal Roll, Pitch, and Yaw Rates (if applicable) from Flight Test Data

   Shows the actual roll, pitch, and yaw rates produced by the vehicle during flight and the roll, pitch, and yaw rates produced by an ideal representation of the inner-loop (first-order transfer function). Note that the roll, pitch, and yaw rate flight commands drive both the actual rates and the ideal rates responses. Displays the performance of the control allocator and dynamic inversion since if these are perfect, then the inner-loop would be equivalent to a first-order transfer function. Small differences in these traces can be attributed to errors produced by the control allocation algorithm and/or the dynamic inversion control law. In flight testing, discrepancies can exist between the model used for dynamic inversion and the actual vehicle and numerous sources of disturbances can exist that contribute to differences between the actual and ideal body rates.

8. Altitude vs. Downrange - (if applicable) from Flight Test Data

The data from simulation was created using batch runs of a non-real-time simulation of the X-40A. The flight test data utilizes raw information from

Figure 3.1: Location and Generation of Command, Actual, and Ideal Roll, Pitch, and Yaw Rates from Simulation.

the actual flight tests. Figure 3.1 shows the location and generation of the command, actual, and ideal body axis rates from the simulation. The flight test data was recorded, however, the ideal roll, pitch, and yaw rates for the flight tests were created by passing the recorded roll, pitch, and yaw rate commands through first-order lag transfer functions.

The commanded and actual body-axis rate plot shows the tracking performance of the inner-loop control system. The commanded body-axis rates are generated by the trajectory/guidance loops. It is desired that the inner-loop track the commanded body-axis rates, hence, a measure of performance is how well the actual rates match the commanded rates. The ideal roll, pitch, and yaw rates correspond to an ideal inner-loop. As was shown in Equation 2.35, the ideal inner-loop looks like a first-order system, with band-widths of $5\frac{rad}{sec}$ for the roll and yaw channels and $3\frac{rad}{sec}$ for the pitch channel. Hence, to test performance of the inner-loop, the commanded roll, pitch, and yaw rates can be filtered by $\frac{5}{s+5}$ or $\frac{3}{s+3}$ (depending on the channel of inter-

est) and the output of this filter is what is termed the ideal body-axis rates. Comparing the ideal rates with the actual rates provides a measure of the performance of the dynamic inversion and control allocation algorithms. Of course, this is only valid when the commands are feasible. Another plot of interest is the inner-loop bandwidth. Recall, that if one or more axes becomes saturated, the inner-loop bandwidth is adjusted. This plot will show the bandwidth that is passed to the guidance loops. The remaining plots show the trajectory followed and the control effector deflections. Not all plots are applicable to all failure cases investigated, so only those which apply will be displayed.

In terms of flight test data, plots of altitude vs. downrange (flight) and flight actual and flight ideal body-axis rates will be shown. In this case, the flight actual body-axis rates are those from the simulation onboard the TIFS aircraft during an approach and landing flight. Here, the onboard simulation is driven by real-world measurements, winds, turbulence, etc. and provides a more realistic result as compared to the desktop simulation studies. The flight ideal body-axis rates are generated using the body-axis rate commands from the onboard simulation and passing these through first-order filters of the form $\frac{5}{s+5}$ for the roll and yaw channels and $\frac{3}{s+3}$ for the pitch channel. Once again, the body-axis rate commands are from the flight data. It is important to make sure that the inner-loop is behaving as desired, hence, the metric in this case is that the flight actual body-axis rates behave similar to the flight ideal body-axis rates. Another metric is the altitude vs. downrange (flight) plots. Here, a quick glance will suffice to validate that the system is behaving as desired.

Another important point to note is that the body-axis rates from the desktop simulation do not equal the body-axis rates from the flight tests. In the flight tests, eight parameters were selected from the TIFS to be used by the X-40A flight control system [see Volume 5]:

1. Three earth referenced velocity components

2. Three earth referenced positions

3. Heading

4. Altitude

Since only these parameters are used by the X-40A model, they are the only quantities that are matched by the TIFS. Mismatches in other quan-

34

tities (roll, pitch, and yaw rates) have minimal impact on the simulation. Hence, one would not expect the desktop simulation and TIFS flight body-axis rates to compare favorably. The correct comparison is the one shown in the plots, that is, separately comparing actual and ideal body-axis rates from the simulation and from the flight tests. Likewise, a comparison between the simulated altitude profile and the flight tested altitude profile cannot be made. Since the TIFS is not capable of matching the drag characteristics of the X-40A, it was necessary to scale the altitude and altitude rates to 30% of the X-40A model (see Volume 5). Hence, even though the altitude profiles should look similar, careful inspection will show that the precise values are not equal, which is to be expected. More information on the model following and vertical motion scaling can be found in Volume 5.

3.1.1 Nominal Results

The results from a nominal run of the simulation will be presented. In this case, all control surfaces are operating normally and the nominal flight path is followed. The first two plots, Figures 3.2 and 3.3 show the roll, pitch, and yaw rates. In Figure 3.2, the commanded and actual rates are displayed and the tracking performance is acceptable. Figure 3.3 shows the actual and ideal body-axis rates. Here, it is easily seen that the inner-loop response appears to track the response of the ideal first-order lag as the traces in these plots are very similar.

Control deflections, from the simulation, are shown in Figures 3.4 and 3.5. Figures 3.6 and 3.7 display the altitude, crossrange, and downrange from the simulation. Performance is acceptable as the nominal flight path is well behaved and the deviation of crossrange from zero is minimal.

The point of showing nominal runs is so that a comparison can be made between these plots and the results of simulation runs involving failure cases. Thus, a nominal set of conditions (flight path, control deflections, body-axis rates) have been defined.

The last two figures in this section display the nominal flight tests results from the TIFS aircraft. Figures 3.8 and 3.9 display the altitude vs. downrange and actual and ideal body-axis rates from a flight test with no failures. Clearly, the altitude profile behaves as expected and the flight actual body-axis rates are nearly identical to the flight ideal body-axis rates. Hence, it is evident that the inner-loop, during this particular flight test, was working properly.
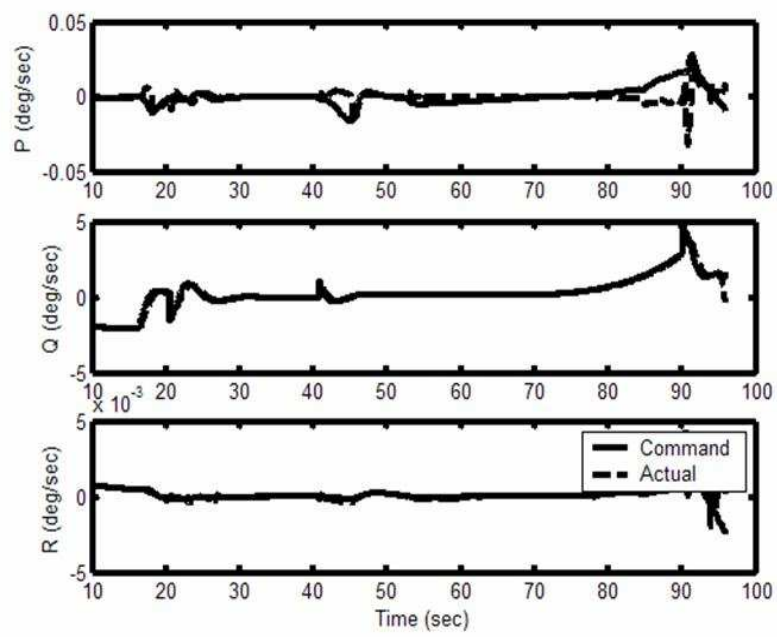
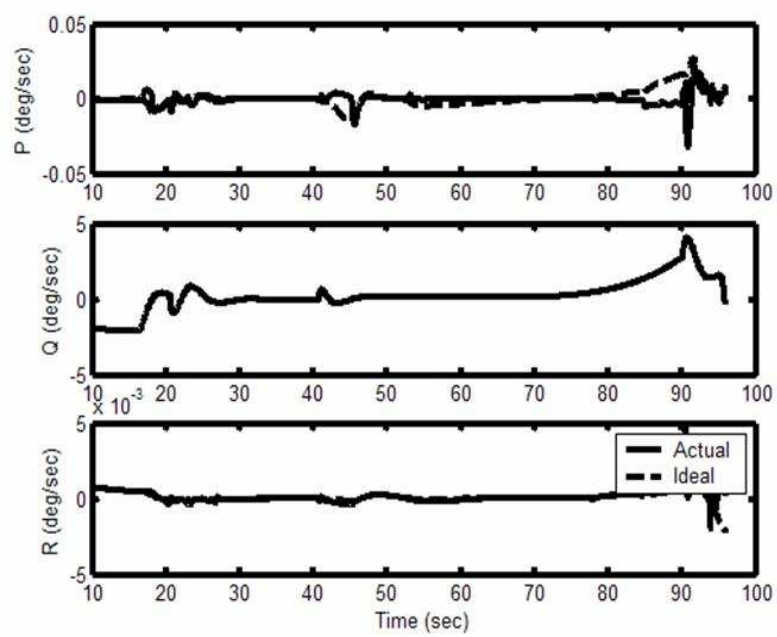Figure 3.2: Command and actual body-axis rates - Simulation - nominal.

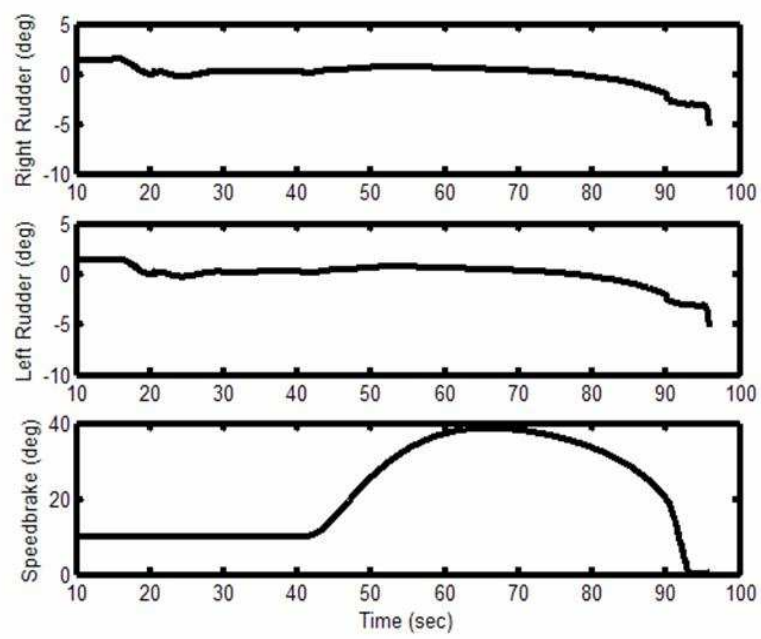Figure 3.3: Actual and ideal body-axis rates - Simulation - nominal.

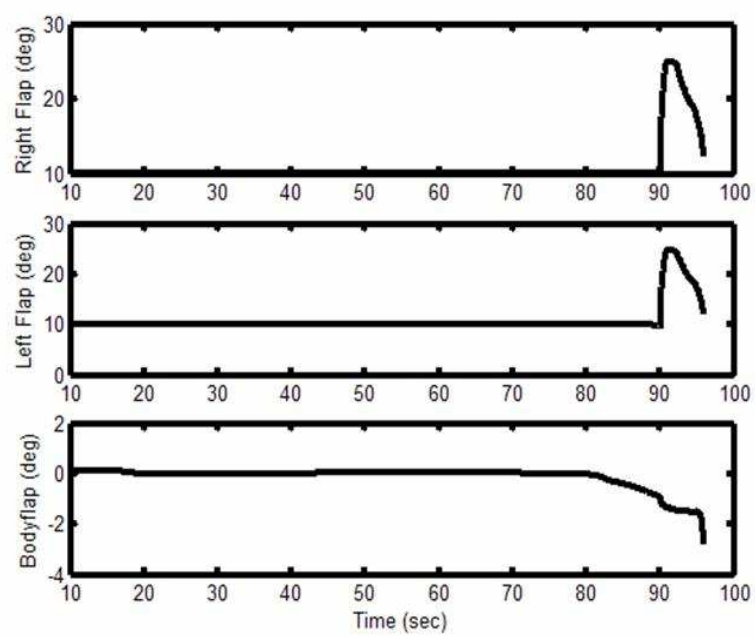Figure 3.4: Control Deflections - Simulation - nominal.

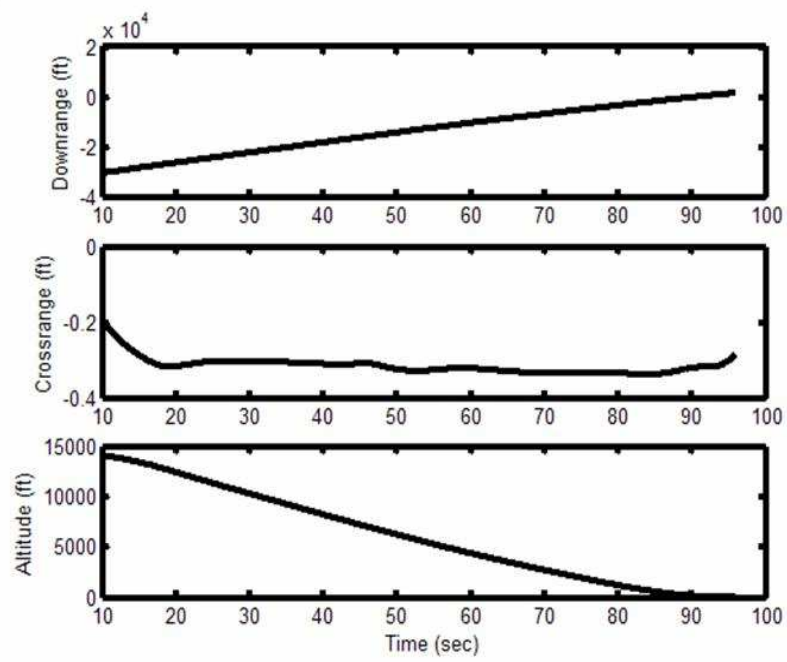Figure 3.5: Control Deflections - Simulation - nominal.

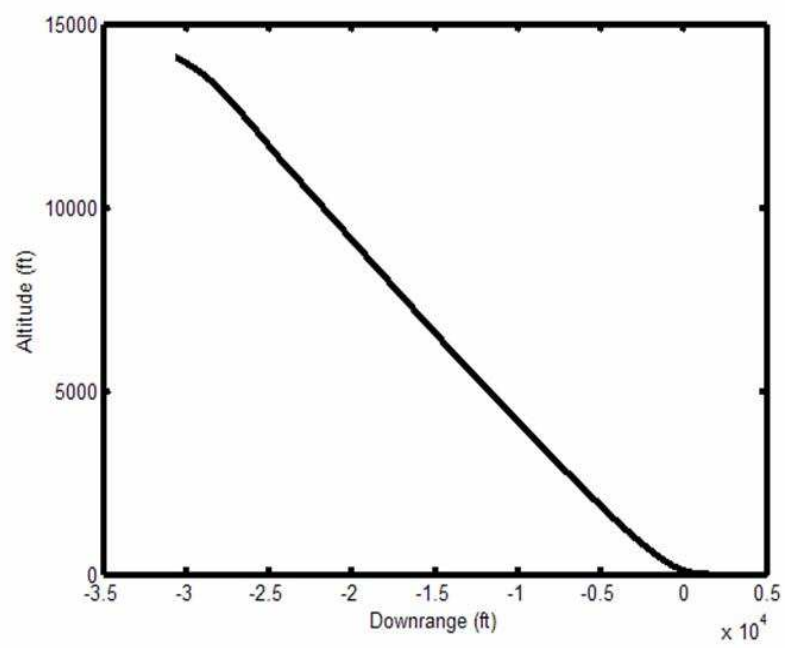Figure 3.6: Altitude, Crossrange, Downrange - Simulation - nominal.

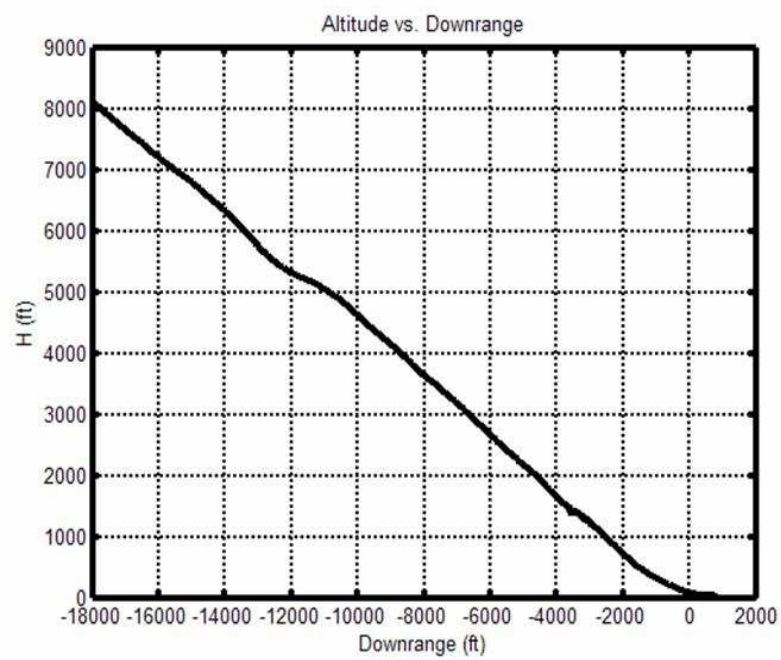Figure 3.7: Altitude vs. Downrange - Simulation - nominal.

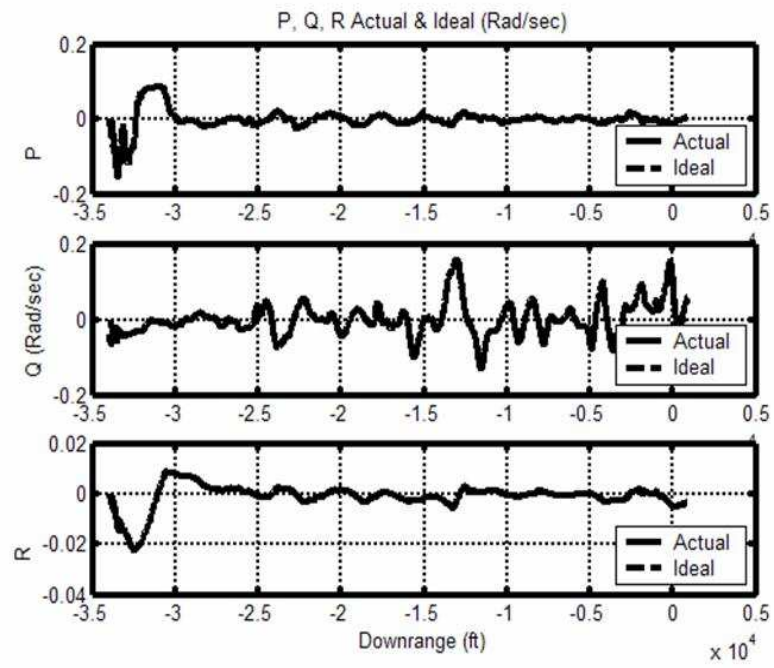Figure 3.8: Altitude vs. Downrange - Flight Data - nominal.

Figure 3.9: Actual and ideal body-axis rates - Flight Data - nominal.

3.1.2 Right Flaperon Failure Results

In this section, results obtained from simulating a right flaperon failure at 15°
will be discussed. This failure results in a large unbalanced rolling moment
if no corrective action is taken (and some unbalanced yawing moment). One
way to overcome these adverse moments is to move the left flaperon to 15°
and use the ruddervators (and bodyflap and speedbrake to a lesser extent)
to control the pitching moment. This is exactly how the IAG&C controller
automatically re-mixes the control effectors after the failure. Without an
onboard reconfigurable inner-loop, the failed right flaperon causes and un-
controllable rolling moment which is fatal to the vehicle. To illustrate these
points, the batch simulation was performed twice resulting in two sets of
plots. The first set is with the IAG&C reconfigurable inner-loop controller.
The second set is with a non-reconfigurable controller. This will ultimately
show some of the benefits of reconfigurable inner-loop control. Note that this
failure was not flight tested and therefore no flight data will be shown.

Right Flaperon Failure Results: IAG&C Controller

Recall that a right flaperon failure at 15° is simulated in this run. Figures 3.10
and 3.11 show the body-axis rates for this simulation run. Here, it can be
seen that the actual rates track the commands and that the inner-loop control
system does behave like a decoupled system of first-order lags.

Control deflections are shown in Figures 3.12 and 3.13. It can be seen
that about 41 sec. into the run, a right flaperon failure at 15° occurs. As
was stated earlier, this failure causes an adverse rolling moment, which the
control allocator overcomes by moving the left flaperon to 15°. Other than a
short transient period, the reconfigurable controller is capable of recovering
from this type of failure, as can be seen in the altitude, crossrange, and
downrange plots shown in Figure 3.14 and 3.15. The altitude vs. downrange
plot is nearly the same as the nominal (non-failed) case, while the crossrange
error is minimal as the deviation is less than 0.5 ft from zero.

Figure 3.16 shows the inner-loop bandwidth. Clearly, the bandwidths
of each axis are constant at the nominal values; therefore, none of the axes
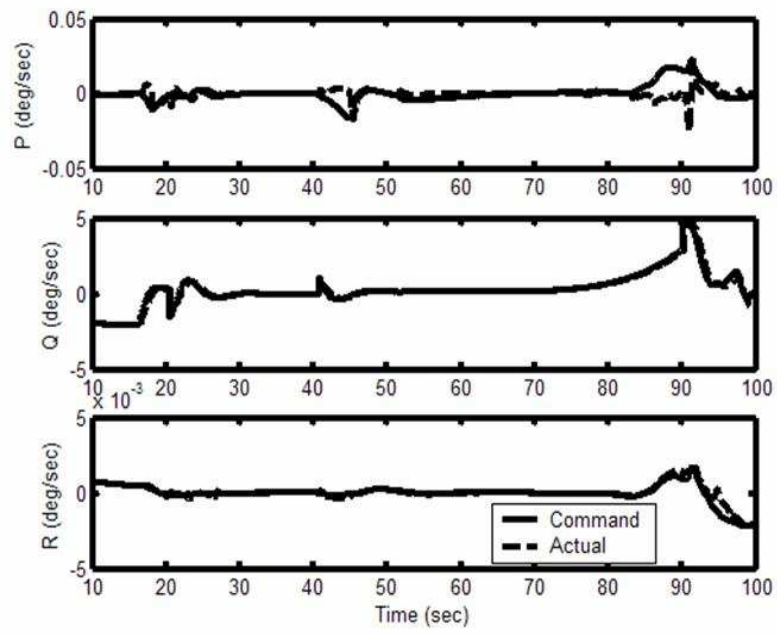become saturated as a result of this failure.

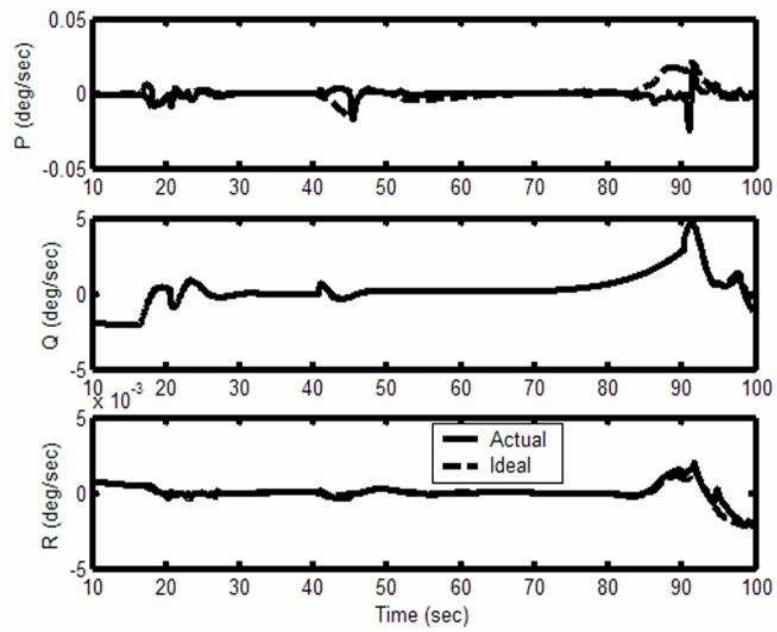Figure 3.10: Command and actual body-axis rates - Right Flaperon = 15°, IAG&C controller - Simulation.

Figure 3.11: Actual and ideal body-axis rates - Right Flaperon = 15°, IAG&C controller - Simulation.

46
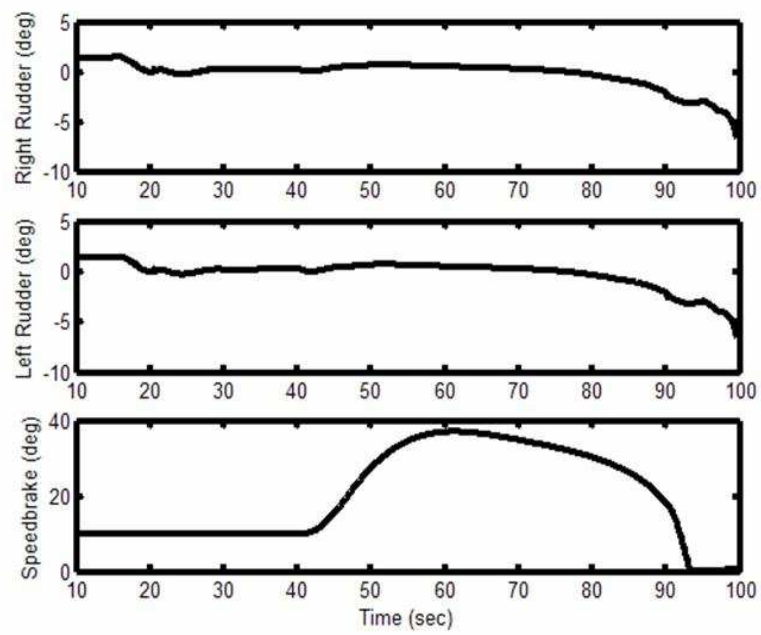
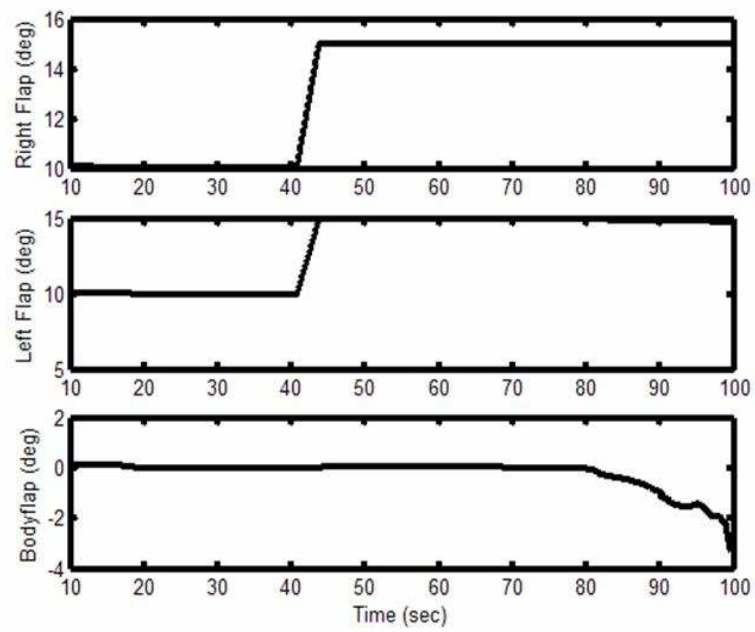Figure 3.12: Control Deflections - Right Flaperon = 15°, IAG&C controller - Simulation.

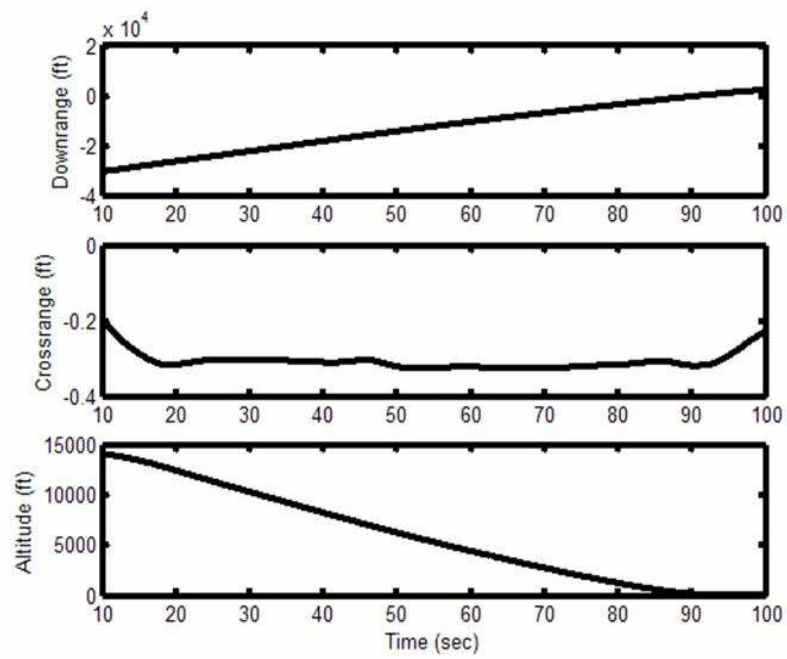Figure 3.13: Control Deflections - Right Flaperon = 15°, IAG&C controller - Simulation.

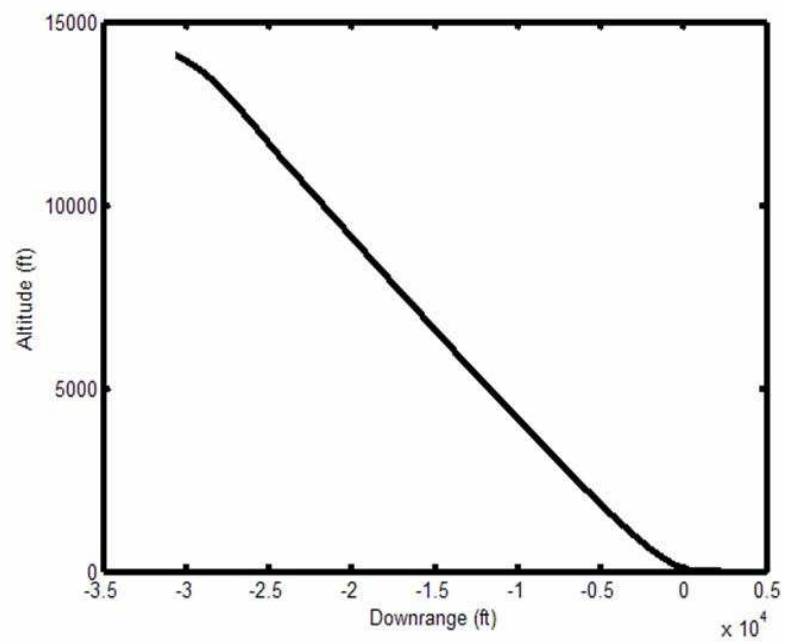Figure 3.14: Altitude, Crossrange, Downrange - Right Flaperon = 15°, IAG&C controller - Simulation.

Figure 3.15: Altitude vs. Downrange - Right Flaperon = 15°, IAG&C controller - Simulation.
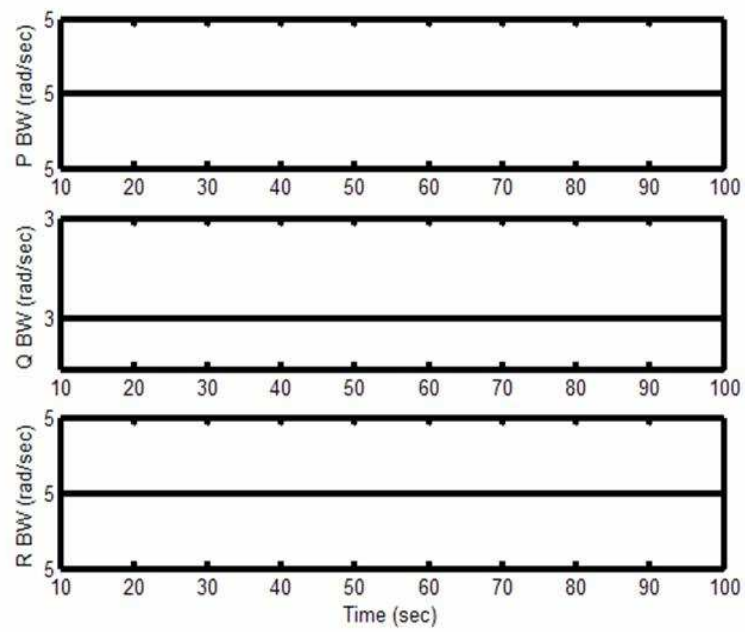
Figure 3.16: Bandwidth - Right Flaperon = 15°, IAG&C controller - Simulation.
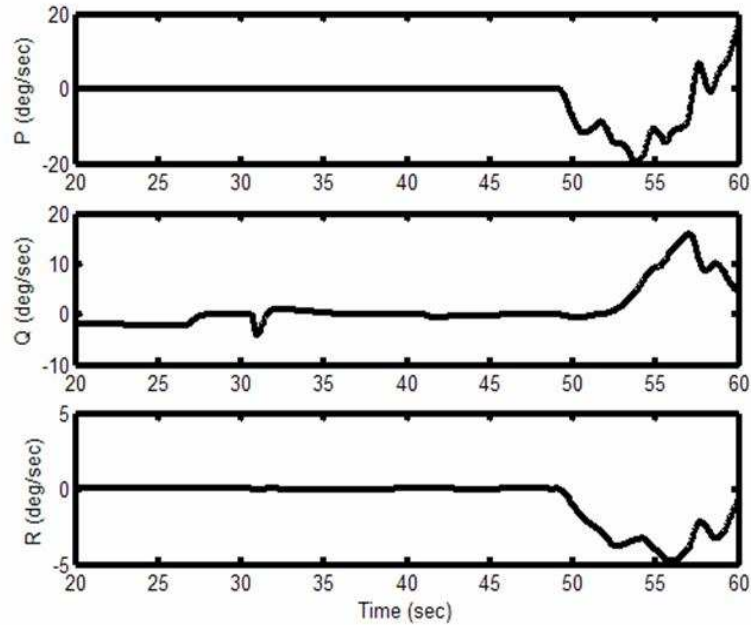
Figure 3.17: Command and actual body-axis rates - Right Flaperon = 15°, Non-reconfigurable controller - Simulation.

Right Flaperon Failure Results: Nominal Controller

Now, the nominal (X-40A baseline non-reconfigurable controller used in the original 8 drop tests of the vehicle [15]) controller is implemented in the simulation. Once again, the right flaperon failure causes an undesired rolling moment; in this case, without reconfiguration capabilities, the system cannot recover from the failure, as can be seen in the body-axis rate plot shown in Figure 3.17. Clearly, the body-axis rates are going unstable and do not resemble the nominal (unfailed) system behavior.

The control deflections are shown in Figures 3.18 and 3.19. It's obvious that some of the available surfaces become saturated and there is not sufficient control authority to follow the nominal flight path.

These same types of issues can be seen in the altitude, crossrange, and downrange plots. Figures 3.20 and 3.21 show this data. Clearly, the crossrange has large deviations from zero and the altitude vs. downrange plot is
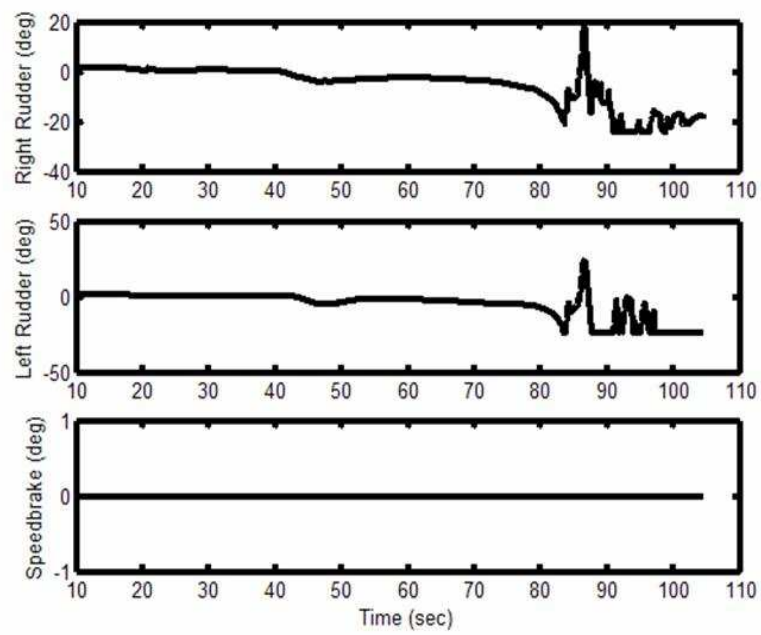
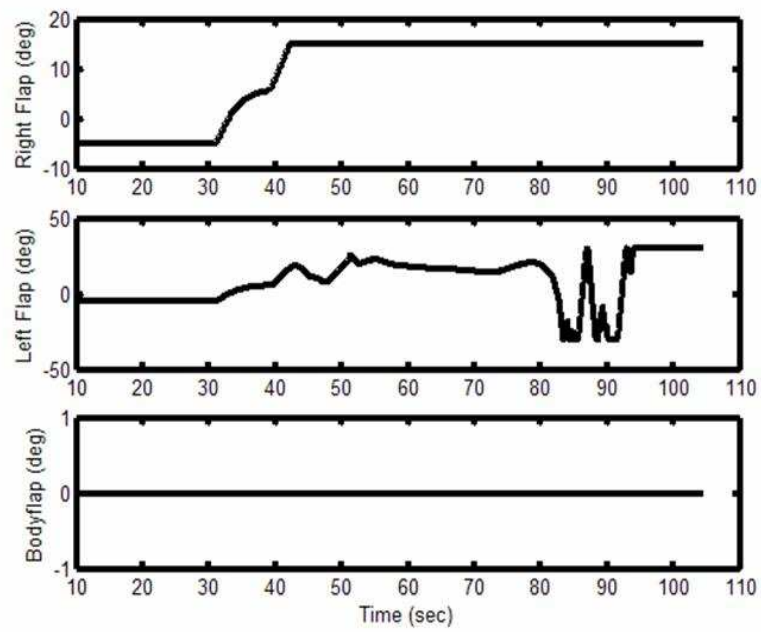Figure 3.18: Control Deflections - Right Flaperon = 15°, Non-reconfigurable controller - Simulation.

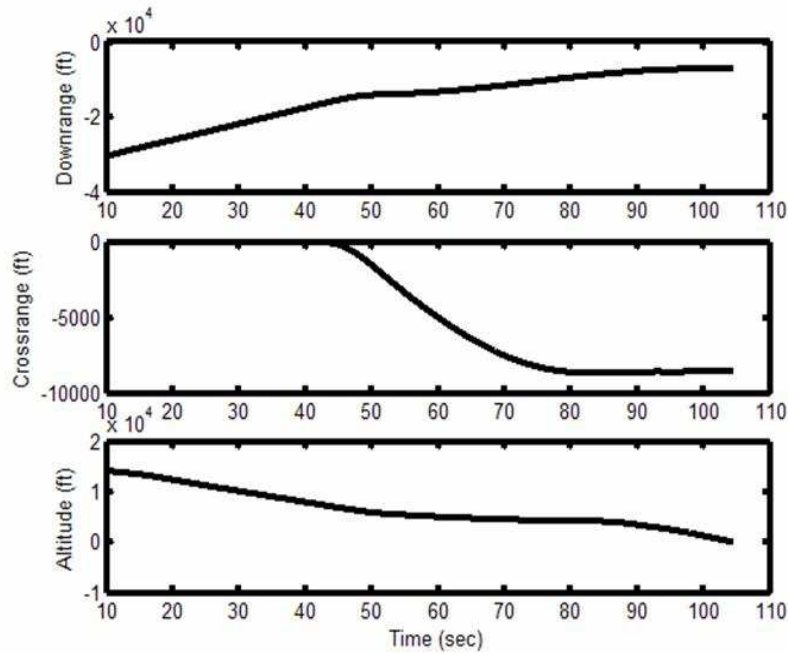Figure 3.19: Control Deflections - Right Flaperon = 15°, Non-reconfigurable controller - Simulation.

Figure 3.20: Altitude, Crossrange, Downrange - Right Flaperon = 15°, Non-reconfigurable controller - Simulation.

not suitable for a successful landing.

The idea behind this comparison is to show that improved performance can be achieved by employing a reconfigurable inner-loop controller. It should be noted that the X-40A baseline controller (non-reconfigurable) does not have access to the bodyflap and speedbrake, hence, it has limited pitching moment capabilities. Fortunately, these two surfaces do not provide and rolling or yawing moments (so they cannot be used to counteract the effects of the failed right flaperon). If the IAG&C reconfigurable inner-loop controller did not have the additional bodyflap and speedbrake, similar acceptable performance (similar to the 6 control surface vehicle with the IAG&C controller) would still be observed. The left flaperon would move to 15° to counter the failed right flaperon, but now the rudders would be required to deflect more to produce the required pitching moments (since the bodyflap and speedbrake produce some pitching moment). In effect, even though the simulations compared the six vs. the four control effector vehicle, the results
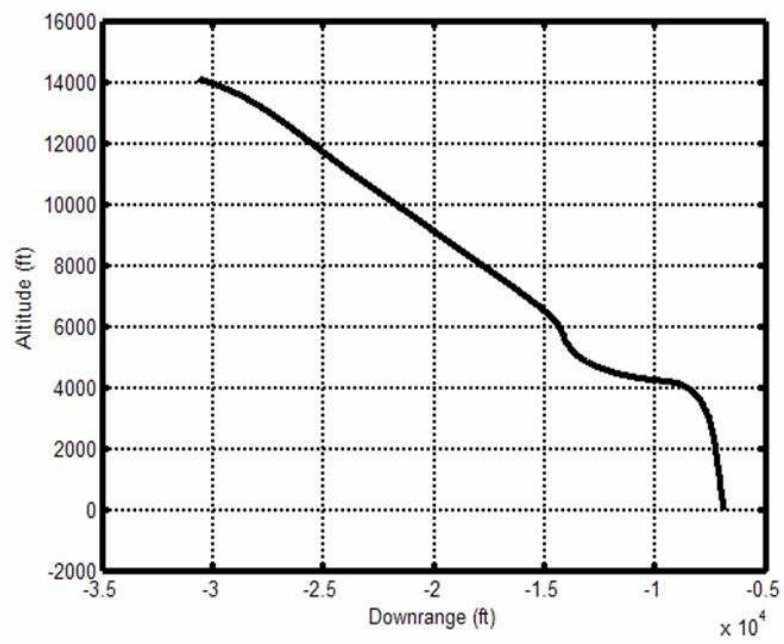
Figure 3.21: Altitude vs. Downrange - Right Flaperon = 15°, Non-reconfigurable controller - Simulation.
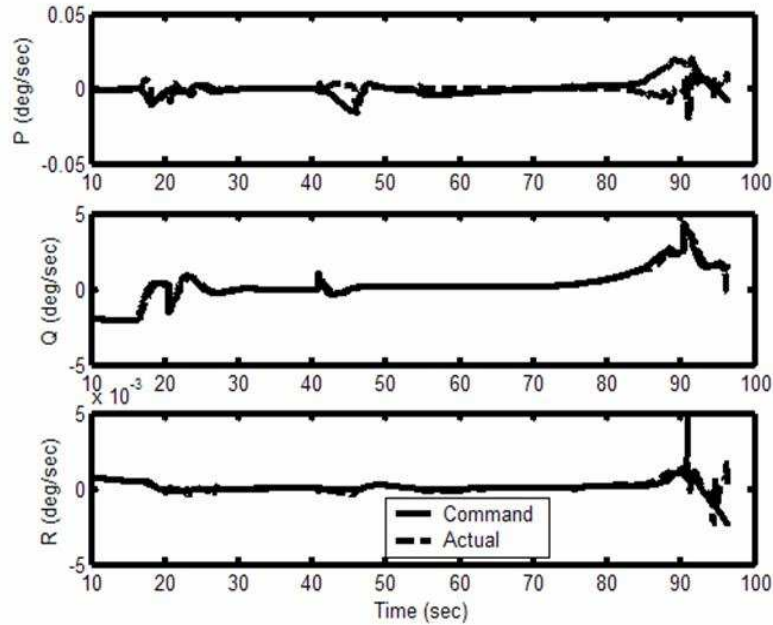
Figure 3.22: Command and actual body-axis rates - Bodyflap = 5°, IAG&C controller - Simulation.

display the benefit of reconfigurable control.

3.1.3 Bodyflap Failure Results

Another failure case investigated is a locked bodyflap at 5°. This type of failure produces similar results to the right flaperon failed at 15°. The IAG&C controller is capable of recovering from this failure with minimal deviation from nominal performance. Figures 3.22 - 3.28 show the plots from this simulation run. Figures 3.29 and 3.30 display the flight test data for a flight with a bodyflap failure at 5°. Once again, the altitude trace is reasonable and the flight actual and flight ideal body-axis rates are nearly equivalent. Hence, the inner-loop is working as expected.
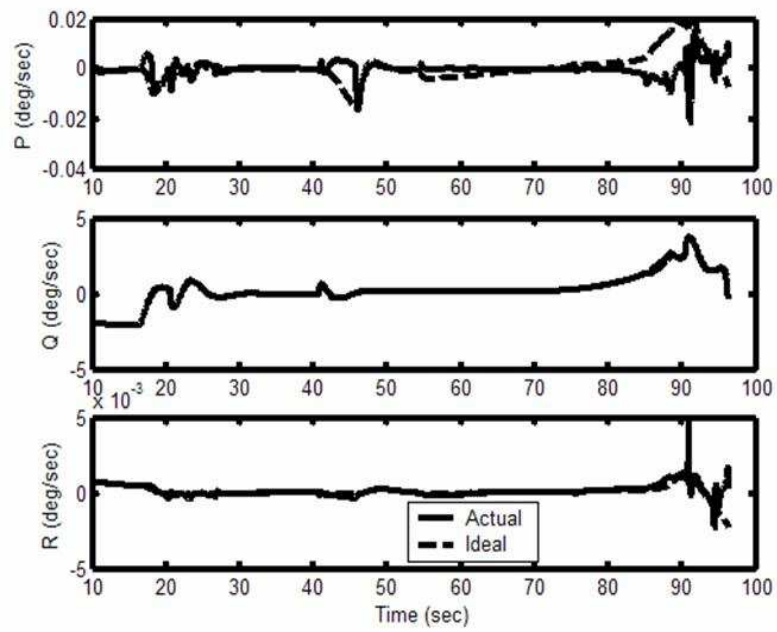
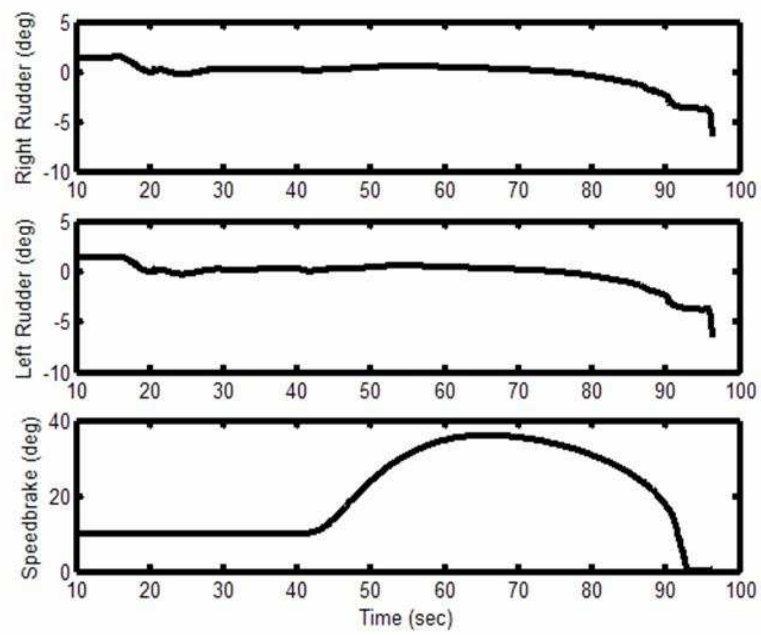Figure 3.23: Actual and ideal body-axis rates - Bodyflap = 5°, IAG&C controller - Simulation.

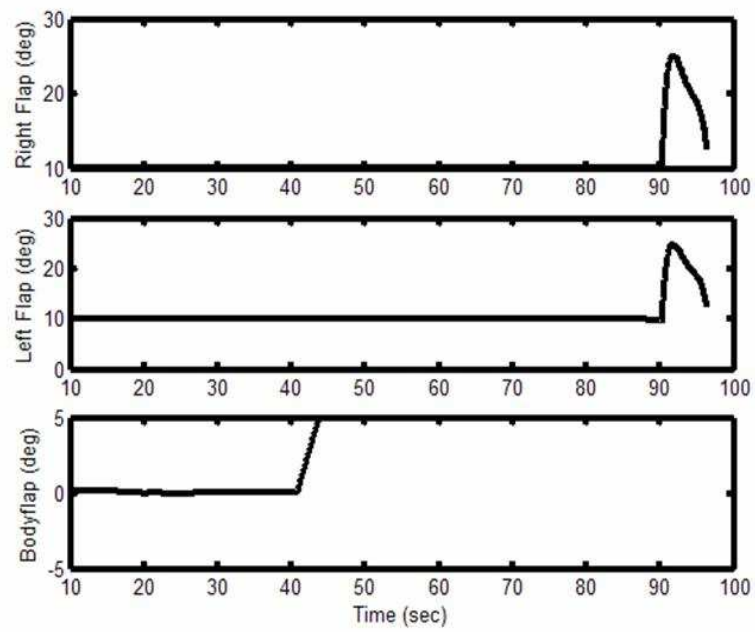Figure 3.24: Control Deflections - Bodyflap = 5°, IAG&C controller - Simulation.

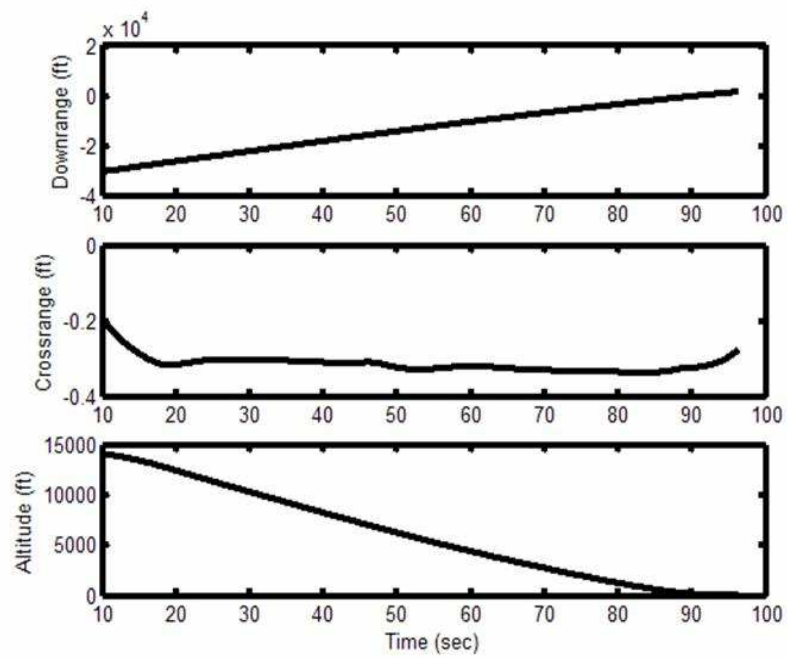Figure 3.25: Control Deflections - Bodyflap = 5°, IAG&C controller - Simulation.

Figure 3.26: Altitude, Crossrange, Downrange - Bodyflap = 5°, IAG&C controller - Simulation.
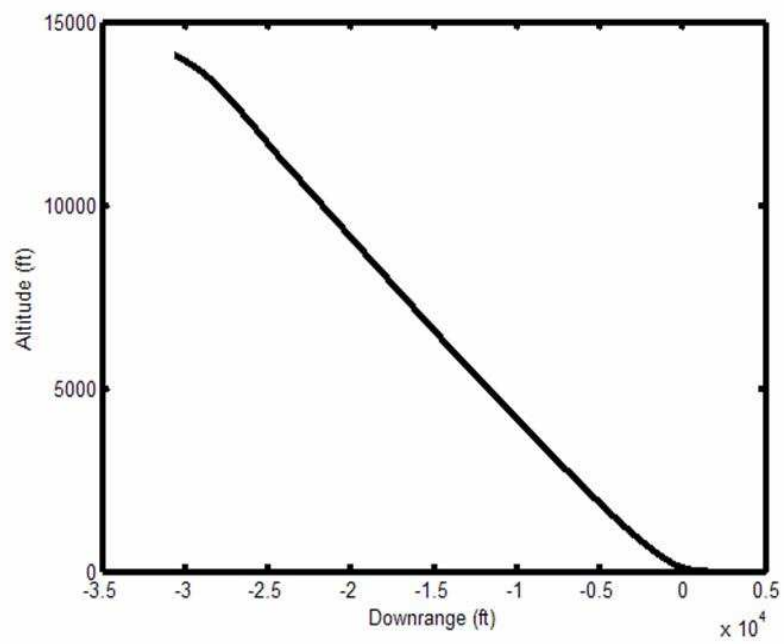
Figure 3.27: Altitude vs. Downrange - Bodyflap = 5°, IAG&C controller - Simulation.
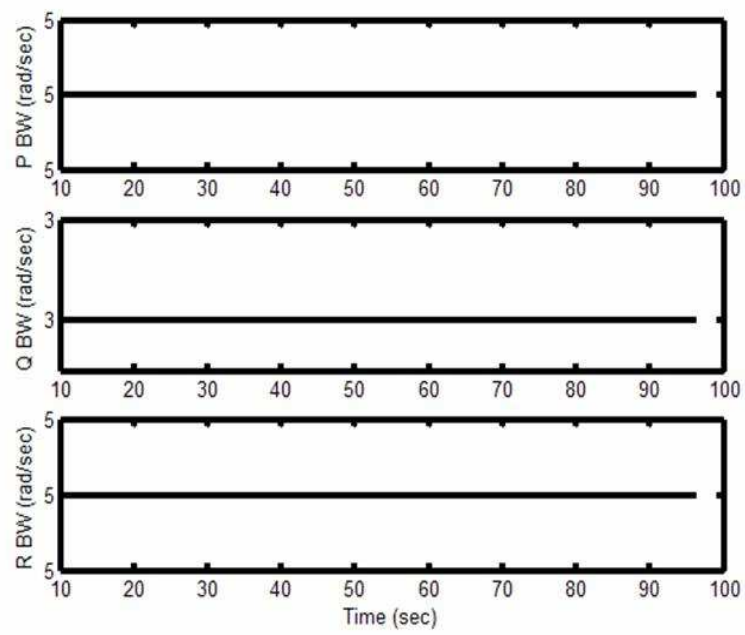
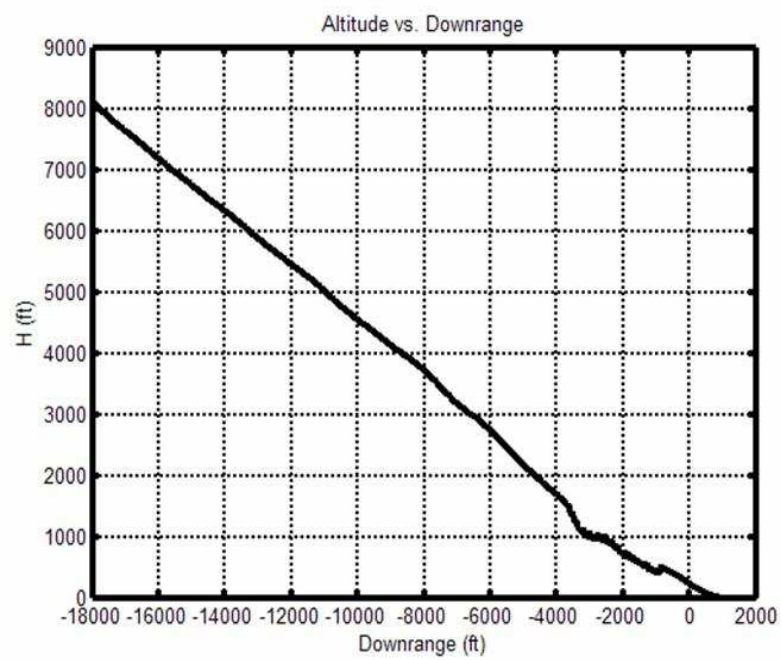Figure 3.28: Bandwidth - Bodyflap = 5°, IAG&C controller - Simulation.

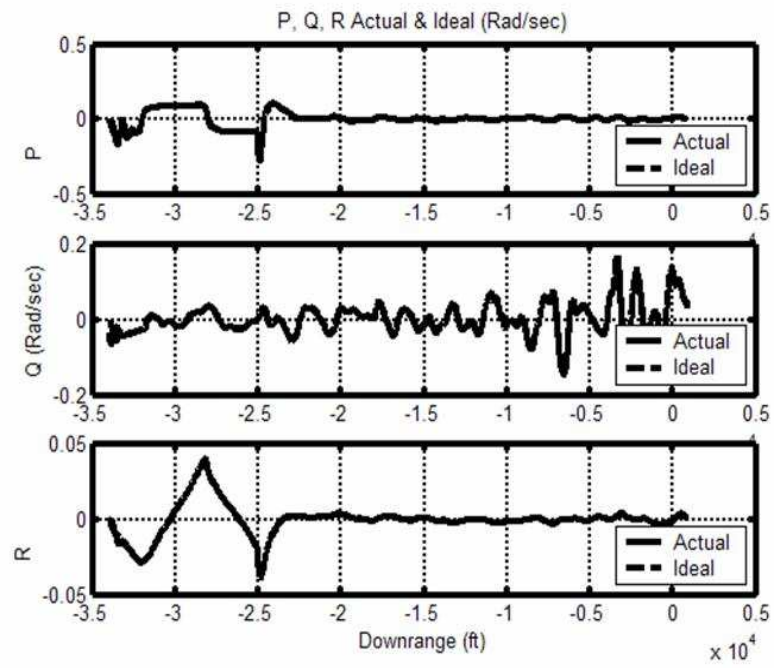Figure 3.29: Altitude vs. Downrange - Bodyflap = 5° - Flight Data.

Figure 3.30: Actual and ideal body-axis rates - Bodyflap = 5° - Flight Data.

3.1.4   Right & Left Ruddervators Failure Results

The last failure case discussed in this report is a combination of two failed control surfaces. Here, both ruddervators are failed at 0°. This is a severe failure for this vehicle as the ruddervators are the most powerful control effectors. Fortunately, there is sufficient control redundancy to recover most of the nominal performance. The flaperons are available to control the roll axis and to some extent the yaw axis, while the bodyflap and speedbrake are used for pitch control.

Figures 3.31 and 3.32 show the body-axis rates for this failure. Once again, performance is acceptable, however, note that there is some error in the yaw axis. Shortly, the cause of this error will be shown. The control effector deflections are shown in Figures 3.33 and 3.34. These figures show that the ruddervators have been locked at 0° and they show that the deflections of the bodyflap and speedbrake are larger than the unfailed case to account for the loss of the ruddervators.

Figures 3.35 and 3.36 display the altitude, crossrange, and downrange. Again, the reconfigurable controller is capable of recovering the vehicle with only a slight deviation from nominal conditions. Figure 3.37 shows the bandwidth of each axis as reported by the inner-loop reconfigurable controller. For the first time, the bandwidth is not constant at its nominal value. In fact, after the failure occurs (41 sec. into the run), the bandwidth of the yaw axis is reduced which indicates that this axis has become saturated. This is why the error is present in the yaw rate shown in Figure 3.31. When the axis saturates, the inner-loop is essentially slowed by reducing the bandwidth and this information is passed to the guidance loops. When guidance gain adaptation is available, the reduced bandwidth from the inner-loop would notify the guidance algorithm to adjust its gains so as to maintain stability. The guidance adaptation algorithm is described in Volume 2.

In conclusion, the failure of both ruddervators at 0° is a failure from which the reconfigurable controller can recover the majority of nominal performance. Some axis saturation occurs, but it is not severe enough to cause large deviations from the nominal flight path.

In terms of flight test data for this failure, Figures 3.38 and 3.39 display the data. The altitude vs. downrange plot is similar to the simulated version so it appears that the entire system is working properly. In terms of isolating the inner-loop, from Figure 3.39, it can be seen that the flight actual and flight ideal body-axis rates are quite close and hence the inner-loop is behaving like
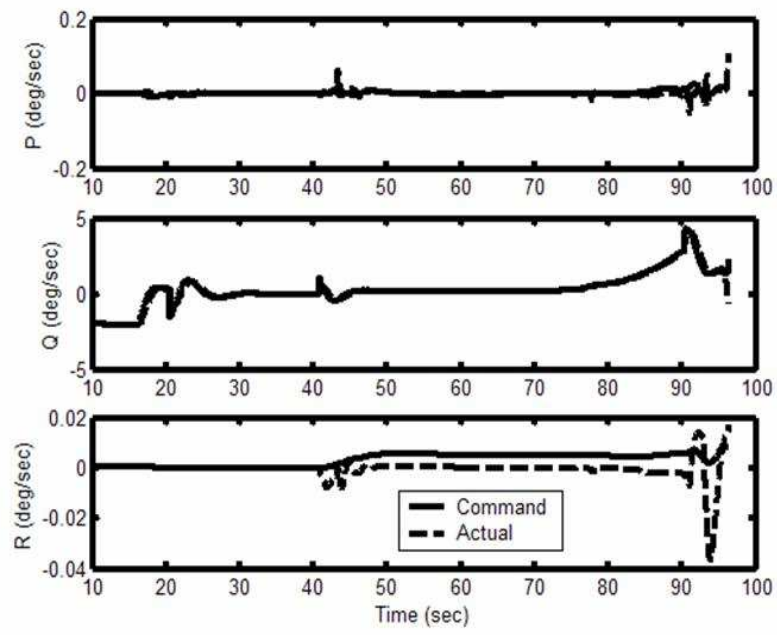
Figure 3.31: Command and actual body-axis rates - Right and Left Rudder-vators = 0°, IAG&C controller - Simulation.
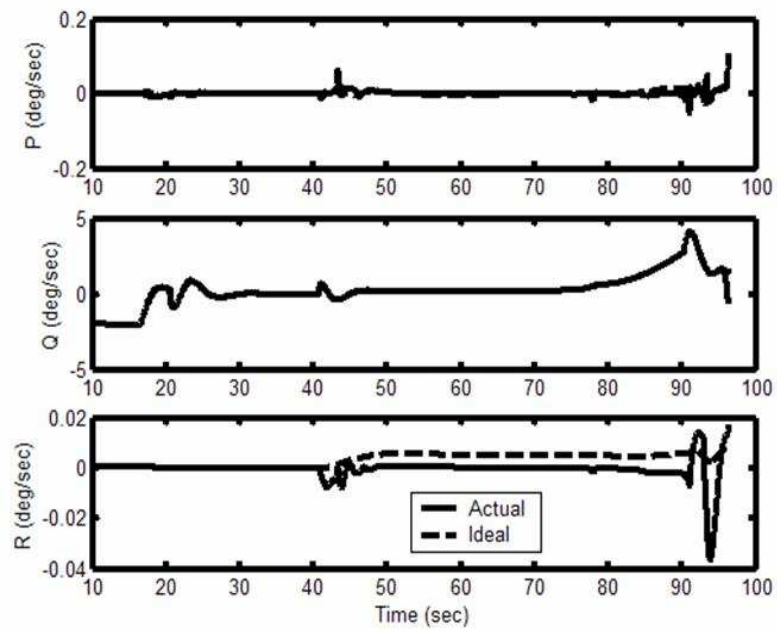
Figure 3.32: Actual and ideal body-axis rates - Right and Left Ruddervators = 0°, IAG&C controller - Simulation.
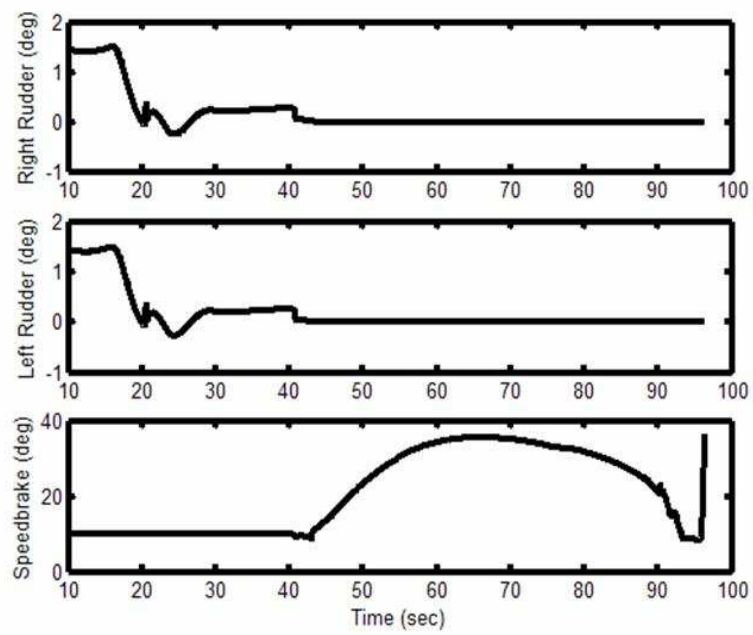
Figure 3.33: Control Deflections - Right and Left Ruddervators = 0°, IAG&C controller - Simulation.
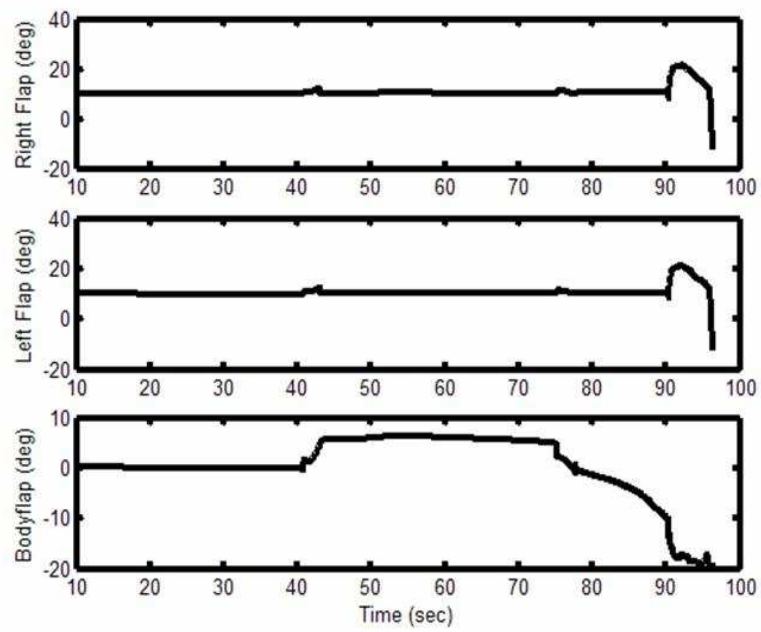
Figure 3.34: Control Deflections - Right and Left Ruddervators = 0°, IAG&C controller - Simulation.
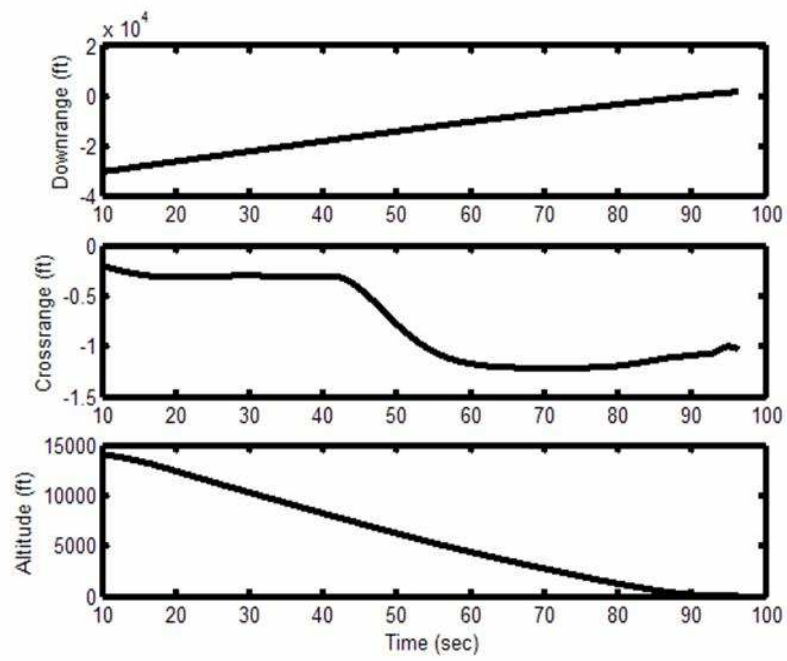
Figure 3.35: Altitude, Crossrange, Downrange - Right and Left Ruddervators = 0°, IAG&C controller - Simulation.

a bank of decoupled first-order lag transfer functions.

Figure 3.36: Altitude vs. Downrange - Right and Left Ruddervators = $0°$, IAG&C controller - Simulation.

Figure 3.37: Bandwidth - Right and Left Ruddervators = 0°, IAG&C controller - Simulation.

73

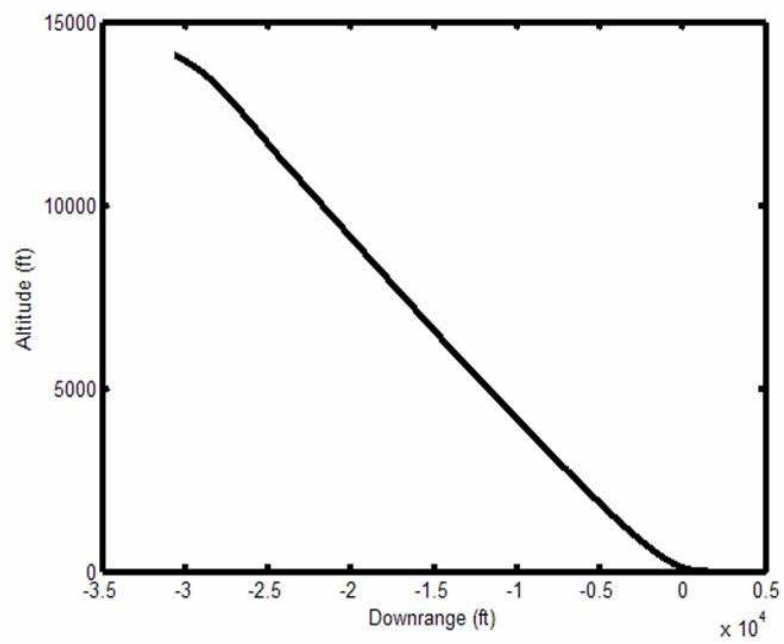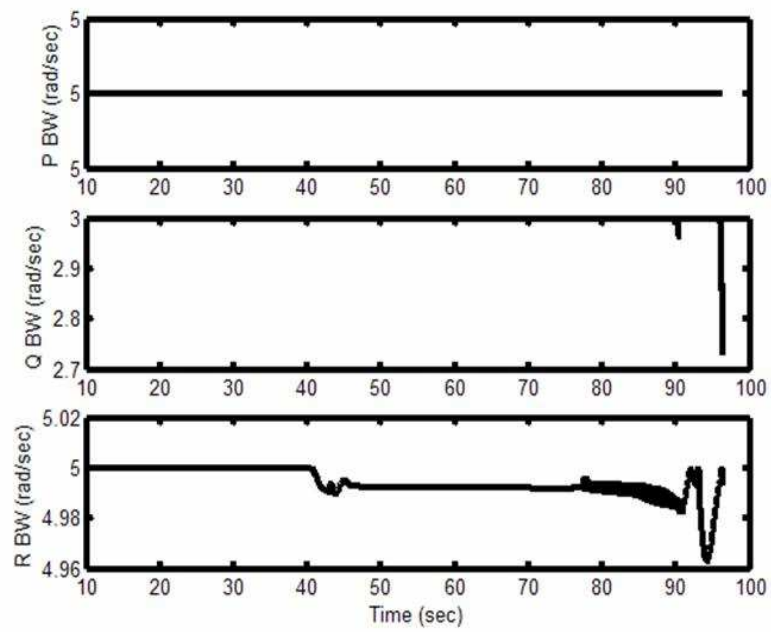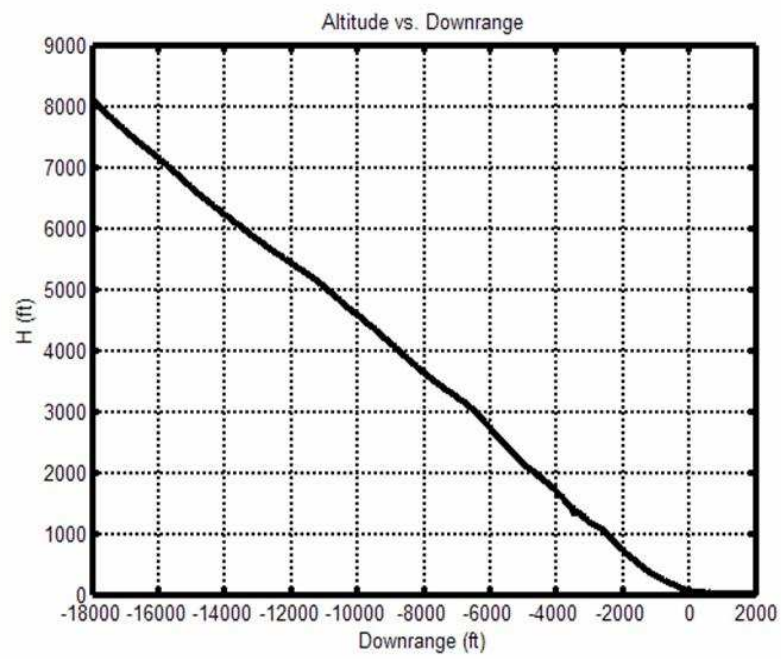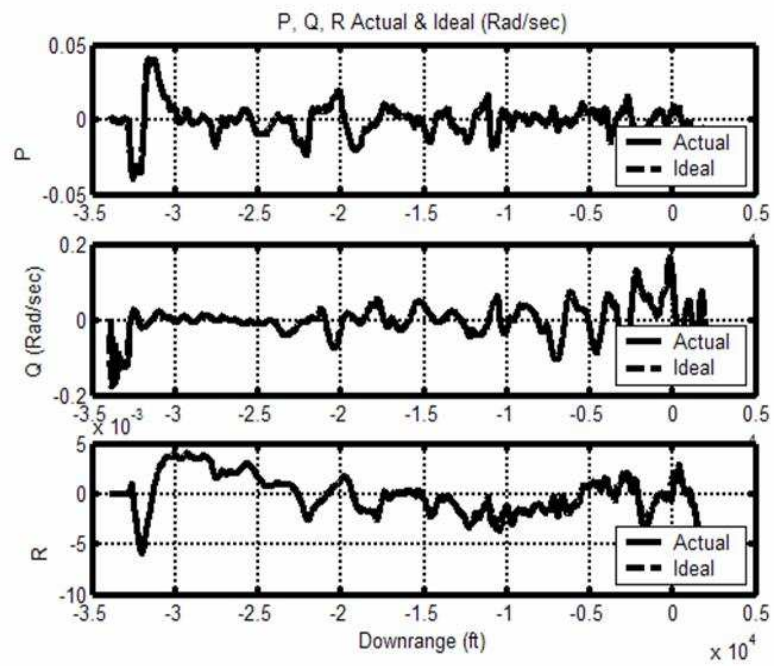Figure 3.38: Altitude vs. Downrange - Right and Left Ruddervators = 0° - Flight Data.

Figure 3.39: Actual and ideal body-axis rates - Right and Left Ruddervators = 0° - Flight Data.

Chapter 4

Conclusions

4.1

The inner-loop reconfigurable control system described in this volume was designed to interact with the adaptive guidance and trajectory reshaping algorithms described in Volume 2. Representative non-real-time and real-time in-flight simulation results that focused on the performance of the inner-loop flight control system were shown in this volume; however, a more comprehensive and general set of results are shown in Volume 3 of this report. The results presented in this volume show that in some cases, a reconfigurable control system can enable recovery from a control effector failure without guidance adaptation or trajectory reshaping. It is also shown that there are some failures that require all three elements of the IAG&C system in order to recover the vehicle. Real-time operation of the inner-loop reconfigurable control system was successfully demonstrated on the Total In-flight Simulator using a 233MHz Power PC. The linear programming based control allocator was the most computationally intense element of the system; however, it was capable of running well within the .02 sec (50Hz) update rate of the FCS. To our knowledge, this is the first time that a linear programming based control allocator has been flight demonstrated. The element of the inner-loop system that consumed the most memory resources was the onboard aircraft model that was used to store the stability and control data used by the dynamic inversion control law and control allocator.

The inner-loop flight control system was exercised in 75 separate approaches and landings (most to virtual runways above ground level). The

system demonstrated the ability to satisfactorily track the body axis rate commands generated by the guidance system in all cases where sufficient control power was available. The reference model bandwidth attenuation algorithm was exercised on several flights and was typically triggered by surface rate limits when the surfaces were responding to tracking errors caused by gusts or turbulence. Due to cost and schedule constraints, some flights were conducted under wind conditions that exceeded the placard wind limits of the X-40A; however, the TIFS aircraft operated well within it's placard limits at all times.

Bibliography

[1] J. S. Brinker and K. A. Wise, "Flight testing of reconfigurable control law on the x-36 tailless aircraft," *Journal of Guidance, Control and Dynamics*, vol. 24, no. 5, pp. 903–909, 2001.

[2] D. G. Ward, J. F. Monaco, and M. Bodson, "Development and flight testing of a parameter identification algorithm for reconfigurable control," *Journal of Guidance, Control and Dynamics*, vol. 21, no. 6, pp. 948–956, 1998.

[3] M. Pachter, P. Chandler, and M. Mears, "Reconfigurable tracking control with saturation," *Journal of Guidance, Control and Dynamics*, vol. 18, no. 5, pp. 1016–1022, 1995.

[4] P. Chandler, M. Pachter, and M. Mears, "System identification for adaptive and reconfigurable control," *Journal of Guidance, Control and Dynamics*, vol. 18, no. 3, pp. 516–522, 1995.

[5] A. J. Calise, S. Lee, and M. Sharma, "Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft," in *Proceedings of the 1998 Guidance, Navigation and Control Conference*, AIAA 1998-4108, August 1998.

[6] "Application of Multivariable Control Theory to Aircraft Control Laws," Tech. Rep. WL-TR-96-3099, Wright Laboratory, WPAFB, OH, 1996.

[7] A. B. Page and M. L. Steinberg, "A Closed-loop Comparison of Control Allocation Methods," in *Proceedings of the 2000 Guidance, Navigation and Control Conference*, AIAA 2000-4538, August 2000.

[8] M. Bodson, "Evaluation of optimization methods for control allocation," *Journal of Guidance, Control and Dynamics*, vol. 25, no. 4, pp. 703–711, 2002.

[9] D. B. Doman and A. D. Ngo, "Dynamic inversion-based adaptive/reconfigurable control of the x-33 on ascent," *Journal of Guidance, Control and Dynamics*, vol. 25, no. 2, pp. 275–284, 2002.

[10] B. Etkin, *Dynamics of Atmospheric Flight*. John Wiley & Sons, Inc., 1972.

[11] D. B. Doman and M. W. Oppenheimer, "Improving Control Allocation Accuracy for Nonlinear Aircraft Dynamics," in *Proceedings of the 2002 Guidance, Navigation and Control Conference*, AIAA 2002-4667, August 2002.

[12] J. M. Buffington, " Modular Control Law Design for the Innovative Control Effectors (ICE) Tailless Fighter Aircraft Configuration 101-3," Tech. Rep. AFRL-VA-WP-TR-1999, Air Force Research Laboratory, Wright-Patterson A.F.B., OH, 1999, pp 93-94.

[13] J. M. Buffington, P. Chandler, and M. Pachter, " Integration of Online System Identification and Optimization-based Control Allocation," Tech. Rep. AIAA Paper 98-4487, American Institute of Aeronautics and Astronautics, 1998.

[14] M. Bolender and D. B. Doman, " Non-linear Control Allocation Using Piecewise Linear Functions," in *Proceedings of the 2003 Guidance, Navigation and Control Conference*, AIAA 2003-5357, August 2003.

[15] R. T. Cervisi, "Mini-Spaceplane Technology Study," Tech. Rep. AFRL-VA-WP-TR-99-1999-3010, Wright Laboratory, WPAFB, OH, 1998.

Appendix A

On Velocity Tracking

As mentioned in Volume 3, both the non-real-time Monte Carlo analysis and TIFS simulation of the X-40A indicated that satisfactory sink rates, touchdown points, terminal pitch attitudes, and touchdown velocities were not achieved on each and every approach and landing. These cases were generally associated with large wind perturbations and so it is expected that performance may have been improved if a speed loop were implemented. While a speed loop was not implemented in the flight demonstration, one concept was explored early in the inner-loop design.

The closed loop concept that was explored was to close the velocity loop by adding a fourth channel to the dynamic inversion based explicit model following system. This involved subtracting the wing-body induced linear acceleration (essentially -D/m) from the desired linear acceleration generated by an error compensating prefilter to form the input to a fourth channel on the control allocator. The control allocator therefore had to find the appropriate distribution of control effector deflections required to generate a set of commanded body axis angular accelerations and forward (total linear acceleration vector) linear acceleration. The angular acceleration error minimization problem posed in Equation 2.11 can be modified to include linear acceleration as follows:

$$\min_{\delta} J_E = \min_{\delta} \mathbf{W_v} \left| \mathbf{B_v} \boldsymbol{\delta} - \mathbf{d_{vdes}} \right|_1 \tag{A.1}$$

subject to

$$\overline{\boldsymbol{\delta}} \leq \boldsymbol{\delta} \leq \underline{\boldsymbol{\delta}} \tag{A.2}$$

where

$$\mathbf{B_v} = \begin{bmatrix} \mathbf{I^{-1}} & 0 \\ \mathbf{0} & 1/m \end{bmatrix} \begin{bmatrix} \frac{\partial L}{\partial \delta_1} \frac{\partial L}{\partial \delta_2} \cdots \frac{\partial L}{\partial \delta_n} \\ \\ \frac{\partial M}{\partial \delta_1} \frac{\partial M}{\partial \delta_2} \cdots \frac{\partial M}{\partial \delta_n} \\ \\ \frac{\partial N}{\partial \delta_1} \frac{\partial N}{\partial \delta_2} \cdots \frac{\partial N}{\partial \delta_n} \\ \\ \frac{\partial D}{\partial \delta_1} \frac{\partial D}{\partial \delta_2} \cdots \frac{\partial D}{\partial \delta_n} \end{bmatrix} \quad (A.3)$$

and $\mathbf{d_{v}}_{des} = [\dot{p}_{des} \quad \dot{q}_{des} \quad \dot{r}_{des} \quad \ddot{x}_{des}]^T$. Notice the inclusion of the weighting vector $\mathbf{W_v}$. Each element of the weighting vector is multiplied by the corresponding element of $|\mathbf{B_v}\boldsymbol{\delta} - \mathbf{d_{vdes}}|_1$. Clearly the top priority is to use the control effectors to maintain control of the rotational degrees of freedom, therefore the relative weighting on the linear acceleration error should always be much smaller than the weighting on the angular acceleration errors. It was found that under nominal conditions, tight velocity tracking could be achieved using this closed loop approach while maintaining good rotational tracking performance. Under failure conditions; however, the velocity tracking performance suffered.

Independent control of four degrees of freedom was difficult when one or more control effectors failed. Furthermore, the control of linear acceleration using control surfaces in an unpowered approach is problematic because of the extremely limited ability of most surfaces (excluding the speedbrake) to generate drag forces. Control surfaces can be used to correct for small linear acceleration errors but large errors are better corrected by manipulating the angle of attack of the wing-body. In short, the task of velocity error correction may be split between the control surfaces and the wing-body; however, the wing-body exerts the most influence and should be used if possible. Manipulation of the wing-body angle of attack is an outer-loop guidance function, while the generation of the reference flight path is a trajectory generation function.

The system tested in the TIFS flights did not include a closed-inner loop velocity tracking system. The original X-40A, which had only flaperons and rudders, deployed the flaperons at $10°$ trailing edge down during the approach and landing phase. The modified X-40A model that was simulated by the TIFS aircraft had a speedbrake and bodyflap. The speedbrake deflection in the control allocation preference vector was biased such that the drag

produced by the speedbrake was equivalent to that generated by 10° trailing edge down of symmetric flaperon deflection. The speedbrake preference was modulated by a command that was proportional to the velocity error that was calculated by the guidance system. This system is described in Volume 2 of this series of reports.

In summary, although it is mathematically possible to effect limited drag control using only the control effectors, it is difficult to perform during failure cases. Under nominal conditions, the method proposed here can be successfully applied to control a vehicle's velocity. However, when control failures occur, engineering decisions need to be made as to the relative importance of velocity and attitude control. In other words, during a failure, it may not be possible to independently control three body-axes and vehicle velocity. More research on this type of approach to velocity control is warranted.